

Longitudinal data analysis (modern methods)

Advanced modelling with R

Juan R Gonzalez

`juanr.gonzalez@isglobal.org`

BRGE -Bioinformatics Research Group in Epidemiology

ISGlobal -Barcelona Institute for Global Health

<http://brge.isglobal.org>

Modelos modernos para datos longitudinales

- ▶ Datos longitudinales recogen observaciones repetidas de la variable respuesta a lo largo del tiempo, en un mismo individuo
- ▶ El análisis correcto de estos datos contempla que la correlación entre las medidas de cada sujeto es tomada en cuenta
- ▶ A parte de las aproximaciones tradicionales (vistas en la clase anterior), también se puede:
 - ▶ Utilizar *Ecuaciones de Estimación Generalizadas*: GEE
 - ▶ Modelos lineales mixtos

Modelos GEE

- ▶ Modelan la esperanza marginal o poblacional incorporando la correlación entre las observaciones correspondientes a un mismo individuo, y se asume independencia de los individuos
- ▶ Admiten que la variable respuesta siga una distribución distinta a la Gausiana
- ▶ Consideran una ecuación de estimación que se escribe en dos partes: una para modelar los parametros de regresión y la segunda para modelar la correlación
- ▶ Son bastante flexibles ya que el modelo sólo necesita explicitar una función “link”, una función de varianza y una estructura de correlación

Modelos GEE

- ▶ Funcionan bien cuando:
 - ▶ el número de observaciones por sujeto es pequeño y el número de sujetos es grande
 - ▶ se tratan estudios longitudinales donde las medidas siempre se toman en el mismo instante de tiempo para todos los sujetos

Modelos GEE: Formulación

- ▶ Parte sistemática [lo mismo que un GLM]

$$g(E(Y_{ij})) = g(\mu_{ij}) = \beta' X_{ij}$$

donde $i = 1, \dots, n$ y $j = 1, \dots, n_i$, y n denota el número de individuos, y n_i el número de medidas repetidas para el individuo i -ésimo

- ▶ Parte aleatoria

$$V(Y_{ij}) = \nu(\mu_{ij})\phi$$

donde ν es la función de la varianza y ϕ el parámetro de escala

- ▶ Además se tiene que explicitar la estructura de la correlación mediante la *working correlation matrix*, $R(\alpha)$

Modelos GEE

■ Independence,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

■ Exchangeable,

$$\begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix}$$

■ Autoregressive order 1,

$$\begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix}$$

■ Unstructured,

$$\begin{pmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1 \end{pmatrix}$$

Modelos GEE

- ▶ No es necesaria la especificación de un modelo estadístico. Es decir, no es necesario conocer $f(y|\text{parámetros})$. Así, son flexibles, pero:
 - ▶ la estimación de las β 's no tiene porqué se la mejor posible
 - ▶ la inferencia está basada en resultados asintóticos
 - ▶ los métodos de validación son complicados
- ▶ La estimación de los parámetros se puede encontrar en muchos sitios (ver por ejemplo Liang y Zeger, Biometrika, 1986 o Zeger et al, Biometrics, 1988)
- ▶ Si hay datos faltantes (missing) la estimación sólo es correcta si los missing son MCAR (missing completely at Random)

Modelos GEE: estimación de parámetros

La función `geeglm` from the `geepack` es muy similar a la `glm`

```
geeglm(formula, family=gaussian, data, id, zcor=NULL, constr,  
        std.err="san.se")
```

<code>formula</code>	Symbolic description of the model to be fitted
<code>family</code>	Description of the error distribution and link function
<code>data</code>	Optional dataframe
<code>id</code>	Vector that identifies the clusters
<code>zcor</code>	Enter a user defined correlation structure
<code>constr</code>	Working correlation structure: "independence", "exchangeable", "ar1", "unstructured", "userdefined"
<code>std.err</code>	Type of standard error to be calculated. Default "san.se" is the robust (sandwich) estimate; use "jack" for approximate jackknife variance estimate

Figure 2: `geeglm` arguments

Modelos GEE: estimación de parámetros

- For a `geeglm` object returned by `geeglm()`, the functions `drop1()`, `confint()` and `step()` do not apply; however `anova()` does apply.
- The function `esticon()` in the `doBy` package computes and test linear functions of the regression parameters for `lm`, `glm` and `geeglm` objects
- Basic syntax,

```
esticon(obj, cm, beta0, joint.test=FALSE)
```

<code>obj</code>	Model object
<code>cm</code>	Matrix specifying linear functions of the regression parameters (one linear function per row and one column for each parameter)
<code>beta0</code>	Vector of numbers
<code>joint.test</code>	If TRUE joint Wald test of the hypothesis $L\beta = \beta_0$ is made, default is one test for each row, $(L\beta)_i = \beta_{0,i}$

esticon()

- Let $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_p)$ denote the estimated parameters. Also let $k = (k_1, \dots, k_p)$ denote a vector of constants; one row of the matrix for the `cm` argument. Then $c = k^T \beta = k_1 \beta_1 + \dots + k_p \beta_p$.
- `esticon()` calculates the linear combinations of the parameter estimates c , the standard error and the confidence interval
- Specify a value for `beta0` to test $H_0 : c = \text{beta0}$
- If `joint.test=TRUE` then all of the linear combinations are tested jointly

Figure 3: `esticon` function

GEE: estimación de parámetros

ohio dataset: Health effect of air pollution. Children followed for four years, wheeze status (resp: 0-no, 1-yes) recorded annually as well as maternal smoking and age (0 is 9 years old) [**datos en formato largo**]

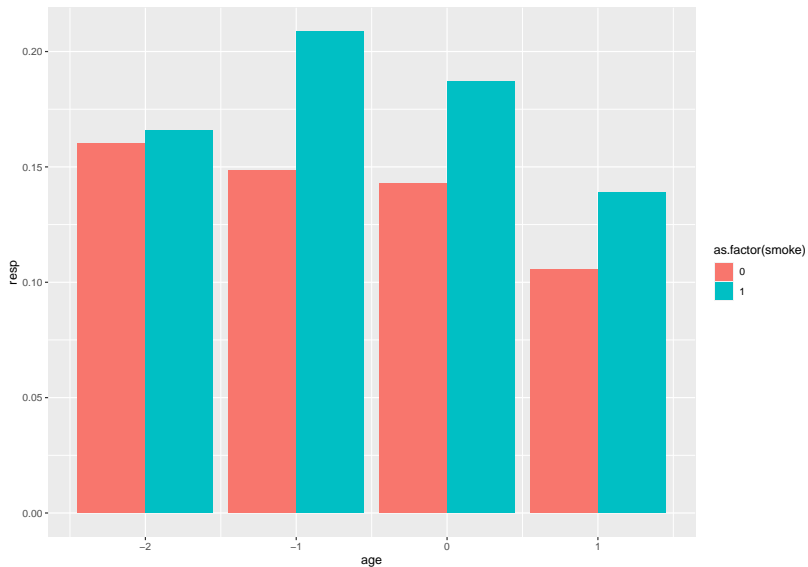
```
library(geepack)
data(ohio)
head(ohio)
```

	resp	id	age	smoke
1	0	0	-2	0
2	0	0	-1	0
3	0	0	0	0
4	0	0	1	0
5	0	1	-2	0
6	0	1	-1	0

GEE: estimación de parámetros

```
library(ggplot2)
dd.agg <- aggregate(resp ~ age + smoke,
                     data=ohio, FUN = mean)
ggplot(dd.agg, aes(y=resp, x=age,
                   fill=as.factor(smoke))) +
  geom_bar(position="dodge", stat="identity")
```

GEE: estimación de parámetros



GEE: estimación de parámetros

El outcome es binario -> binomial

```
fit.exch <- geeglm(resp~age*smoke,  
                  family=binomial(link="logit"),  
                  data=ohio, id=id,  
                  corstr = "exchangeable",  
                  std.err="san.se")  
fit.unstr <- geeglm(resp~age*smoke,  
                   family=binomial(link="logit"),  
                   data=ohio, id=id,  
                   corstr = "unstructured",  
                   std.err="san.se")
```

GEE: estimación de parámetros

```
summary(fit.exch)
```

Call:

```
geeglm(formula = resp ~ age * smoke, family = binomial(link = "logit"),  
       data = ohio, id = id, corstr = "exchangeable", std.err = "san.se")
```

Coefficients:

	Estimate	Std.err	Wald	Pr(> W)
(Intercept)	-1.9005	0.1191	254.69	<2e-16 ***
age	-0.1412	0.0582	5.89	0.015 *
smoke	0.3138	0.1878	2.79	0.095 .
age:smoke	0.0708	0.0883	0.64	0.422

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = exchangeable

Estimated Scale Parameters:

	Estimate	Std.err
(Intercept)	0.999	0.114

Link = identity

Estimated Correlation Parameters:

	Estimate	Std.err
alpha	0.355	0.063

Number of clusters: 537 Maximum cluster size: 4

GEE: estimación de parámetros

```
summary(fit.unstr)
```

Call:

```
geeglm(formula = resp ~ age * smoke, family = binomial(link = "logit"),  
       data = ohio, id = id, corstr = "unstructured", std.err = "san.se")
```

Coefficients:

	Estimate	Std.err	Wald	Pr(> W)
(Intercept)	-1.9084	0.1191	256.61	<2e-16 ***
age	-0.1418	0.0585	5.88	0.015 *
smoke	0.3016	0.1885	2.56	0.110
age:smoke	0.0685	0.0892	0.59	0.443

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = unstructured

Estimated Scale Parameters:

	Estimate	Std.err
(Intercept)	1.01	0.117

Link = identity

Estimated Correlation Parameters:

	Estimate	Std.err
alpha.1:2	0.350	0.0737
alpha.1:3	0.308	0.0715
alpha.1:4	0.304	0.0714
alpha.2:3	0.469	0.0869
alpha.2:4	0.319	0.0741
alpha.3:4	0.378	0.0796

Number of clusters: 537 Maximum cluster size: 4

GEE: estimación de parámetros

Edad como categórica

```
fit <- geeglm(resp~factor(age)+smoke, family=binomial(link="logit"),
              data=ohio, id=id, corstr = "exchangeable",
              std.err="san.se")
summary(fit)
```

Call:

```
geeglm(formula = resp ~ factor(age) + smoke, family = binomial(link = "logit"),
       data = ohio, id = id, corstr = "exchangeable", std.err = "san.se")
```

Coefficients:

	Estimate	Std.err	Wald	Pr(> W)
(Intercept)	-1.7434	0.1374	161.00	<2e-16 ***
factor(age)-1	0.0540	0.1323	0.17	0.68
factor(age)0	-0.0278	0.1388	0.04	0.84
factor(age)1	-0.3755	0.1467	6.55	0.01 *
smoke	0.2712	0.1781	2.32	0.13

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = exchangeable

Estimated Scale Parameters:

	Estimate	Std.err
(Intercept)	1	0.115

Link = identity

Estimated Correlation Parameters:

	Estimate	Std.err
alpha	0.354	0.0636

Number of clusters: 537 Maximum cluster size: 4

GEE: estimación de parámetros

Podemos testar el efecto de una covariable usando un test de razón de verosimilitud

```
fit1 <- geeglm(resp ~ factor(age) + smoke,  
               family=binomial(link="logit"),  
               data=ohio, id=id, corstr = "exchangeable",  
               std.err="san.se")  
fit2 <- geeglm(resp ~ factor(age), family=binomial(link="logit"),  
               data=ohio, id=id, corstr = "exchangeable",  
               std.err="san.se")
```

GEE: estimación de parámetros

```
anova(fit1, fit2)
```

Analysis of 'Wald statistic' Table

Model 1 resp ~ factor(age) + smoke

Model 2 resp ~ factor(age)

	Df	X2	P(> Chi)
1	1	2.32	0.13

GEE: estimación de parámetros

O el test de Wald

```
summary(fit1)
```

Call:

```
geeglm(formula = resp ~ factor(age) + smoke, family = binomial,  
       data = ohio, id = id, corstr = "exchangeable", std.err = TRUE)
```

Coefficients:

	Estimate	Std.err	Wald	Pr(> W)
(Intercept)	-1.7434	0.1374	161.00	<2e-16 ***
factor(age)-1	0.0540	0.1323	0.17	0.68
factor(age)0	-0.0278	0.1388	0.04	0.84
factor(age)1	-0.3755	0.1467	6.55	0.01 *
smoke	0.2712	0.1781	2.32	0.13

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation structure = exchangeable

GEE: estimación de parámetros

Podemos hacer un test para un parámetro en particular. En este caso lo haremos para cada parámetro

```
library(doby)
est <- esticon(fit, diag(5))
rownames(est) <- names(coef(fit))
est
```

	estimate	std.error	statistic	p.value	beta0	df
(Intercept)	-1.7434	0.1374	160.9952	0.0000	0.0000	1
factor(age)-1	0.0540	0.1323	0.1667	0.6831	0.0000	1
factor(age)0	-0.0278	0.1388	0.0400	0.8415	0.0000	1
factor(age)1	-0.3755	0.1467	6.5521	0.0105	0.0000	1
smoke	0.2712	0.1781	2.3191	0.1278	0.0000	1

GEE: estimación de parámetros

ORs e IC95%

```
OR.CI <- exp(cbind(est$estimate, est$lwr, est$upr))  
rownames(OR.CI) <- names(coef(fit))  
colnames(OR.CI) <- c("OR", "Lower OR", "Upper OR")  
OR.CI
```

	OR	Lower OR	Upper OR
(Intercept)	0.175	0.134	0.229
factor(age)-1	1.055	0.814	1.368
factor(age)0	0.973	0.741	1.277
factor(age)1	0.687	0.515	0.916
smoke	1.312	0.925	1.859

GEE: estimación de parámetros

- ▶ Podemos preguntarnos ... ¿cuál es el riesgo de 'wheezing' a los 9 años para un niño cuya madre ha fumado?
- ▶ Es decir, estima $[\text{smoke} + \text{factor}(\text{age})0] - [\text{factor}(\text{age})-1]$

```
val <- esticon(fit, c(0,-1,1,0,1))  
exp(val$estimate)
```

```
[1] 1.21
```

```
val
```

	estimate	std.error	statistic	p.value	beta0	df
[1,]	0.189	0.215	0.773	0.379	0.000	1

Modelos mixtos

Como vimos en la sesión anterior, se podría usar un modelo lineal, pero:

- ▶ Las observaciones repetidas en cada grupo o cluster, no son necesariamente independientes.
- ▶ Con frecuencia, no solo se quieren tomar decisiones respecto de los grupos o cluster observados, sino que se quiere valorar el efecto de las variables explicativas en una población de la que los grupos son una muestra.
- ▶ Puede ser de interés valorar la variación del efecto de x de un grupo a otro.
- ▶ La estimación del efecto medio de las variables explicativas en cada grupo puede ser muy deficiente si no se recoge la posible variabilidad entre los grupos.

Modelos mixtos

- ▶ Modeliza la relación entre la variable dependiente y las covariables
- ▶ Estima la correlación intra-individuo (se puede especificar una estructura)
- ▶ Se pueden aplicar a muchas situaciones (datos multinivel, ANOVA, datos longitudinales)
- ▶ No requieren puntos equidistantes (son covariables -se modeliza el efecto)
- ▶ Son robustos ante los missing

Modelos mixtos

Un modelo mixto se puede representar como:

$$y = X\beta + Zu + \epsilon$$

donde

- ▶ $[y]$ son las observaciones, con media $E(y) = X\beta$
- ▶ $[\beta]$ es un vector de efectos fijos
- ▶ $[u]$ is un vector i.i.d de variables aleatorias con media $E(u) = 0$ y matriz de varianza-covarianza $\text{var}(u) = G$
- ▶ $[\epsilon]$ es un vector de términos i.i.d. correspondientes al error aleatorio con media $E(\epsilon) = 0$ y varianza $\text{var}(\epsilon) = R$
- ▶ $[X \text{ and } Z]$ son matrices de regresores que relacionan las observaciones y con β y u

Modelos mixtos

- Modelo sencillo para interpretar (modelo lineal mixto con intercept aleatorio)

$$y_{ij} = \beta_0 + \beta_1 X_{ij} + a_{ij} + \epsilon_{ij}$$

$$a_i \sim N(0, \tau_a^2), \tau_a^2 \geq 0$$

$$\epsilon_{ij} \sim N(0, \tau^2), \tau^2 > 0$$

- El modelo presenta ahora un intercept aleatorio (centrado en 0) que depende del individuo i -ésimo
- La varianza del efecto aleatorio recoge la variabilidad entre los diferentes individuos
- La varianza del error recoge la variabilidad dentro de cada individuo no explicada por el modelo. NOTA: si la varianza del efecto aleatorio fuese nula, el modelo coincidiría con el modelo de efectos fijos o de regresión lineal.

Modelos mixtos

- ▶ librería nlme o lme4
- ▶ **BodyWeight**: Rat weight over time for different diets

```
library(nlme)
data("BodyWeight" , package="nlme")
head(BodyWeight)
```

Grouped Data: weight ~ Time | Rat

	weight	Time	Rat	Diet
1	240	1	1	1
2	250	8	1	1
3	255	15	1	1
4	260	22	1	1
5	262	29	1	1
6	258	36	1	1

Modelos mixtos

Debemos especificar la estructura de los datos mediante la función `groupedData`

```
datos.s <- groupedData(weight ~ Time | Rat, BodyWeight)
head(datos.s)
```

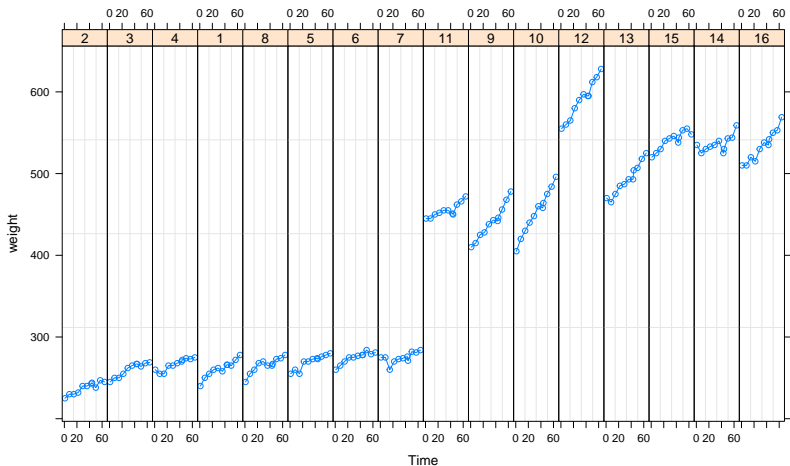
Grouped Data: weight ~ Time | Rat

	weight	Time	Rat	Diet
1	240	1	1	1
2	250	8	1	1
3	255	15	1	1
4	260	22	1	1
5	262	29	1	1
6	258	36	1	1

Modelos mixtos

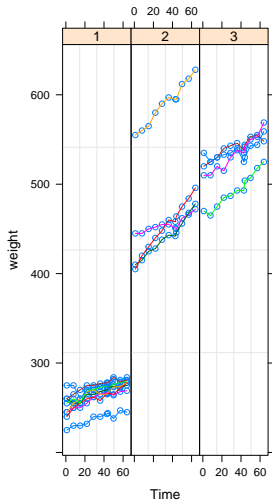
Usa la librería `trellis` para graficar (muy potente)

```
plot(datos.s)
```



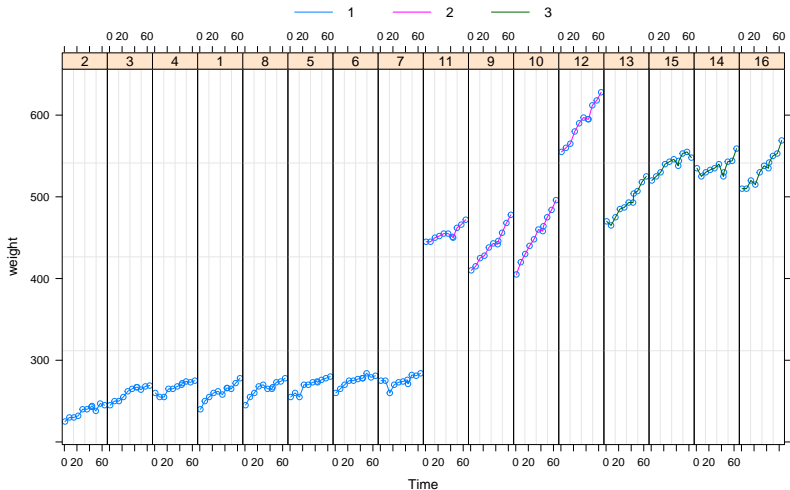
Modelos mixtos

```
plot(datos.s, outer="Diet")
```



Modelos mixtos

```
plot(datos.s, inner="Diet")
```



Modelos mixtos

El modelo de intercept aleatorio puede estimarse con:

```
mod.lme <- lme(weight ~ Time * Diet, datos.s, random = ~ 1)
mod.lme
```

Linear mixed-effects model fit by REML

Data: datos.s

Log-restricted-likelihood: -616

Fixed: weight ~ Time * Diet

(Intercept)	Time	Diet2	Diet3	Time:Diet2	Time:Diet3
251.652	0.360	200.665	252.072	0.606	0.298

Random effects:

Formula: ~1 | Rat

(Intercept) Residual

StdDev: 36.6 6.4

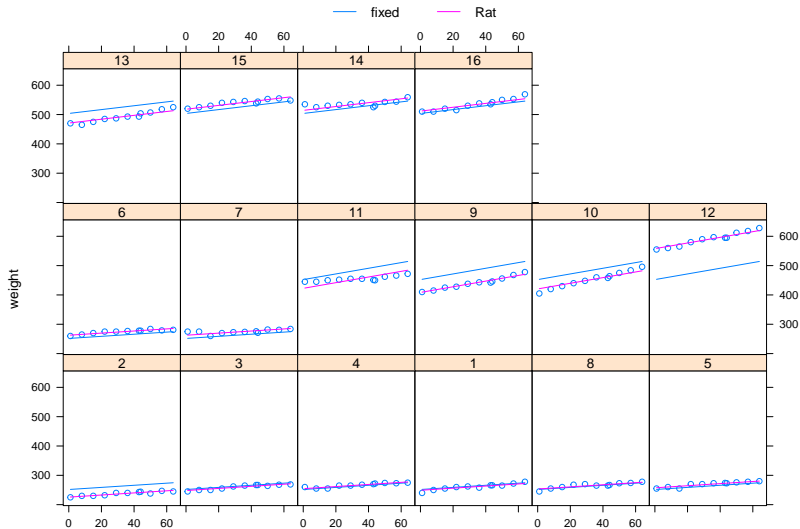
Number of Observations: 176

Number of Groups: 16

Modelos mixtos

Y podemos estimar la parte fija y aleatoria con

```
plot(augPred(mod.lme, level = 0:1, length.out = 2))
```



Modelos mixtos

```
summary(mod.lme)
```

Linear mixed-effects model fit by REML

Data: datos.s

AIC	BIC	logLik
1248	1273	-616

Random effects:

Formula: ~1 | Rat

(Intercept) Residual

StdDev:	36.6	6.4
---------	------	-----

Fixed effects: weight ~ Time * Diet

	Value	Std.Error	DF	t-value	p-value
(Intercept)	251.7	13.01	157	19.34	0
Time	0.4	0.04	157	10.25	0
Diet2	200.7	22.54	13	8.90	0
Diet3	252.1	22.54	13	11.18	0
Time:Diet2	0.6	0.06	157	9.97	0
Time:Diet3	0.3	0.06	157	4.91	0

Correlation:

	(Intr)	Time	Diet2	Diet3	Tm:Dt2
Time		-0.090			
Diet2		-0.577	0.052		
Diet3		-0.577	0.052	0.333	
Time:Diet2		0.052	-0.577	-0.090	-0.030
Time:Diet3		0.052	-0.577	-0.030	-0.090

Standardized Within-Group Residuals:

Min	Q1	Med	Q3	Max
-2.72699	-0.46788	0.00709	0.47622	3.36035

Number of Observations: 176

Number of Groups: 16

Modelos mixtos

Comparamos con un modelo lineal

```
mod.lm <-lm(weight ~ Time * Diet, BodyWeight)
summary(mod.lm)
```

Call:

```
lm(formula = weight ~ Time * Diet, data = BodyWeight)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-51.83	-24.22	0.66	10.44	113.89

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	251.652	7.263	34.65	<2e-16 ***
Time	0.360	0.187	1.92	0.057 .
Diet2	200.665	12.581	15.95	<2e-16 ***
Diet3	252.072	12.581	20.04	<2e-16 ***
Time:Diet2	0.606	0.324	1.87	0.064 .
Time:Diet3	0.298	0.324	0.92	0.359

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34.2 on 170 degrees of freedom

Multiple R-squared: 0.93, Adjusted R-squared: 0.928

F-statistic: 450 on 5 and 170 DF, p-value: <2e-16

Modelos mixtos

El modelo con intercept y pendiente aleatoria puede estimarse con:

```
mod.lme2 <-lme(weight ~ Time * Diet, random = ~ Time,  
              datos.s)  
summary(mod.lme2)
```

Linear mixed-effects model fit by REML

Data: datos.s

	AIC	BIC	logLik
	1172	1203	-576

Random effects:

Formula: ~Time | Rat

Structure: General positive-definite, Log-Cholesky parametrization

	StdDev	Corr
--	--------	------

(Intercept)	36.939	(Intr)
-------------	--------	--------

Time	0.248	-0.149
------	-------	--------

Residual	4.444	
----------	-------	--

Fixed effects: weight ~ Time * Diet

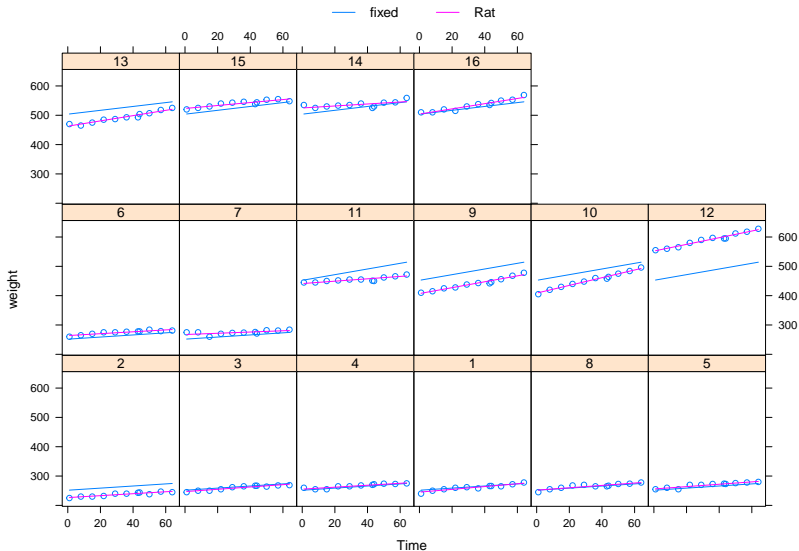
	Value	Std.Error	DF	t-value	p-value
(Intercept)	251.7	13.09	157	19.22	0.0000
Time	0.4	0.09	157	3.95	0.0001
Diet2	200.7	22.68	13	8.85	0.0000
Diet3	252.1	22.68	13	11.11	0.0000
Time:Diet2	0.6	0.16	157	3.84	0.0002
Time:Diet3	0.3	0.16	157	1.89	0.0606

Correlation:

	(Intr)	Time	Diet2	Diet3	Tm:Dt2
Time	-0.160				
Diet2	-0.577	0.092			
Diet3	-0.577	0.092	0.333		
Time:Diet2	0.092	-0.577	-0.160	-0.053	
Time:Diet3	0.092	-0.577	-0.053	-0.160	0.333

Modelos mixtos

```
plot(augPred(mod.lme2, level = 0:1, length.out = 2))
```



Modelos mixtos

¿cuál es necesario?

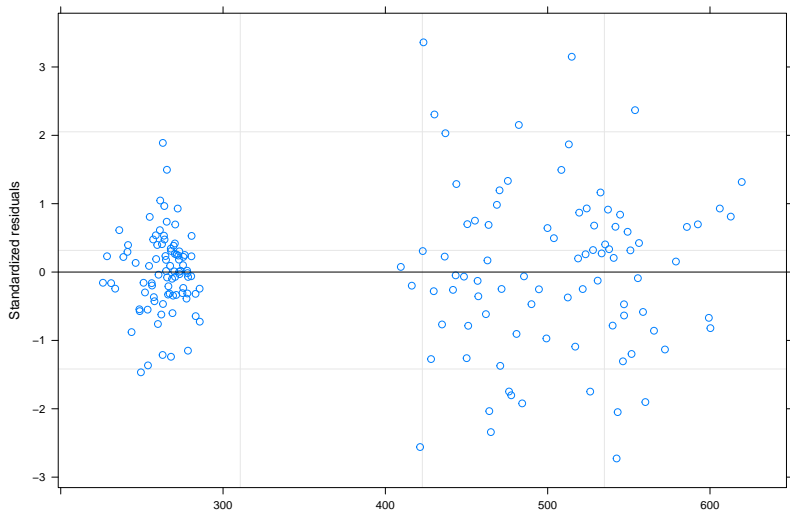
```
anova(mod.lme, mod.lme2)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
mod.lme	1	8	1248	1273	-616			
mod.lme2	2	10	1172	1203	-576	1 vs 2	80.5	<.0001

Modelos mixtos

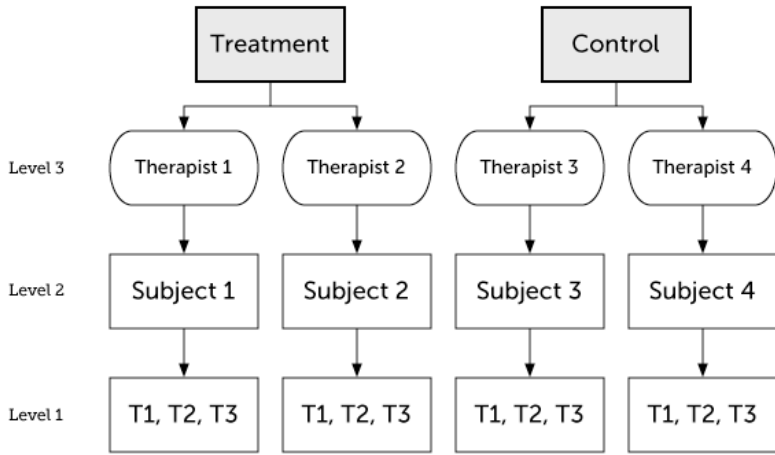
Model checking

```
plot(mod.lme)
```



Modelos anidados

Los **efectos anidados**, el factor aparece SÓLO dentro de un nivel particular de otro factor (cada ojo es de un individuo); para los **efectos cruzados** un factor puede aparecer en más de un nivel de otro factor



Modelos anidados

```
mod.nested <- lme(y ~ time*covar,  
                  random = ~ time | teraphist/subject,  
                  data = df)  
summary(mod.nested)
```

Modelos para datos binarios

ohio dataset: Health effect of air pollution. Children followed for four years, wheeze status (resp: 0-no, 1-yes) recorded annually as well as maternal smoking and age (0 is 9 years old) [**datos en formato largo**]

```
library(lme4)
mod.bin <- glmer(resp ~ age*smoke + (1|id),
                  data=ohio, family="binomial" )
```

Modelos para datos binarios

```
summary(mod.bin)
```

Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']

Family: binomial (logit)

Formula: resp ~ age * smoke + (1 | id)

Data: ohio

AIC	BIC	logLik	deviance	df.resid
1599	1628	-795	1589	2143

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.399	-0.178	-0.159	-0.128	2.602

Random effects:

Groups	Name	Variance	Std.Dev.
id	(Intercept)	5.5	2.35

Number of obs: 2148, groups: id, 537

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	-3.4017	0.2788	-12.20	<2e-16 ***
age	-0.2170	0.0868	-2.50	0.012 *
smoke	0.4782	0.2992	1.60	0.110
age:smoke	0.1046	0.1391	0.75	0.452

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)	age	smoke
age		0.272	
smoke	-0.442		-0.193
age:smoke	-0.146	-0.621	0.280

Modelos para datos binarios

Estimates with 95% CI

```
se <- sqrt(diag(vcov(mod.bin)))  
tab <- cbind(Est = fixef(mod.bin),  
             Lower = fixef(mod.bin) - 1.96 * se,  
             Upper = fixef(mod.bin) + 1.96 * se)
```

Modelos para datos binarios

```
tab
```

	Est	Lower	Upper
(Intercept)	-3.402	-3.948	-2.8552
age	-0.217	-0.387	-0.0469
smoke	0.478	-0.108	1.0648
age:smoke	0.105	-0.168	0.3773

Exercises

1. Analiza de nuevo los datos `agudezavisual.txt` usando un modelo mixto (ojo dentro de individuo) [NOTA: diferente al modelo tradicional que agregaría los datos]. ¿Hay un efecto tiempo?
2. El dataset `Milk` estudia el contenido de proteína en la leche de vaca (variable `protein`). Se dispone de datos desde el parto (variable `Time`) para distintas vacas (variable `Cow`). Estamos interesados en saber qué dieta (variable `Diet`) produce una leche con un mayor contenido de proteínas. Contesta a esta pregunta analizando los datos que puedes cargar con

```
data("Milk", package="nlme")
```

3. Investigadores están interesados en saber si la distancia entre dientes (variable `distance`) en niños y niñas (variable `Sex`) que llevan ortodocia evoluciona de la misma manera a lo largo del tiempo (variable `age`). Contesta a esta pregunta científica analizando los datos que puedes cargar en R con:

```
data("Orthodont", package="nlme")
```