

Análisis de supervivencia y de datos longitudinales (M0-161)

Tercera prueba de evaluación continua - dataset ARMD

María Plaza García - DNI 45735741

11 de Juni, 2020

Índice

1. Introducción	2
2. Datos	2
2.1. Selección de la muestra estudio	2
2.2. Exploración de la muestra estudio	4
2.2.1. ARMD Trial: Agudeza visual	4
2.2.2. Patrones de datos faltantes	4
2.2.3. Perfiles de valor medio	6
2.2.4. Varianzas y correlaciones de las mediciones de agudeza visual	10
3. Modelos lineales mixtos (LMM)	13
3.1. Modelo con intersecciones aleatorias y varianza residual homogénea	13
3.2. Modelo con intercepciones aleatorias y la funcion <code>varPower()</code> , de varianza residual	17
3.2.1. Gráficos de diagnóstico	19
3.3. Modelos con intersecciones y pendientes aleatorias	24
3.3.1. Modelo con una matriz general D	24
3.3.2. Modelo con una Matriz Diagonal D	25
3.3.3. Modelo con una matriz diagonal D y un efecto de tratamiento constante	31
3.4. Una función alternativa de varianza residual: <code>varIdent()</code>	33
3.5. Prueba de hipótesis sobre efectos aleatorios	34
3.5.1. Prueba de intercepciones aleatorias	36
3.5.2. Prueba de pendientes aleatorias	37
3.6. Análisis usando la función <code>lmer()</code>	39
Bibliografía	41

Repositorio Github:

https://github.com/mariaplaza/PEC3_DatosLongitudinales

1. Introducción

Los modelos mixtos son usados cuando los datos tienen algún tipo de estructura jerárquica o de agrupación como los diseños de medidas repetidas, las series temporales, los diseños anidados o bloques aleatorizados. Los modelos mixtos permiten tener coeficientes fijos (aquellos cuyos niveles son de interés) y aleatorios (aquellos cuyos niveles son sólo una realización de todos los posibles niveles procedentes de una población) y varios términos de error. Son una herramienta muy útil, pero también son difíciles de comprender y aplicar.

En esta PEC se va a reproducir el análisis estadístico completo que se lleva a cabo en la selección de Galecki and Burzykowski (2013) para una muestra concreta del dataset ARMD. Desde la presentación y descripción del dataset objeto de estudio hasta el modelo lineal mixto propuesto. Cada paso es razonado y al final se hace una interpretación de los resultados.

Para ello se emplea especialmente el paquete `nlme()` (Pinheiro et al. 2013) y al final se realiza una muestra de los resultados obtenidos con el paquete `lme4` (Bates et al. 2015).

2. Datos

2.1. Selección de la muestra estudio

Vamos a emplear los datos `ARMD.wide` disponibles en el paquete `nlmeU` como nuestro data frame de estudio. Los datos de ARMD surgen de un ensayo clínico aleatorizado multicéntrico que compara un tratamiento experimental (interferón alfa) versus placebo para pacientes diagnosticados con ARMD, de las siglas en inglés, Age-Related Macular Degeneration (La degeneración macular relacionada con la edad). Existen tres conjuntos de datos, con la misma información distribuida de forma diferente, según se precise para el análisis: `armd`, `armd0` y `armd.wide`. La muestra principal la tomaremos de los datos denominados `armd.wide`.

Veamos las características principales de los datos:

```
library(nlmeU)
attach(armd.wide)
attach(armd0)
attach(armd)
dim(armd.wide)
```

```
[1] 240 10
```

```
str(armd.wide)
```

```
'data.frame': 240 obs. of 10 variables:
 $ subject : Factor w/ 240 levels "1","2","3","4",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ lesion : int 3 1 4 2 1 3 1 3 2 1 ...
 $ line0 : int 12 13 8 13 14 12 13 8 12 10 ...
 $ visual0 : int 59 65 40 67 70 59 64 39 59 49 ...
 $ visual4 : int 55 70 40 64 NA 53 68 37 58 51 ...
 $ visual12: int 45 65 37 64 NA 52 74 43 49 71 ...
 $ visual24: int NA 65 17 64 NA 53 72 37 54 71 ...
 $ visual52: int NA 55 NA 68 NA 42 65 37 58 NA ...
 $ treat.f : Factor w/ 2 levels "Placebo","Active": 2 2 1 1 2 2 1 1 2 1 ...
 $ miss.pat: Factor w/ 9 levels "----","---X",...: 4 1 2 1 9 1 1 1 2 ...
```

Vemos que se trata de un data frame con datos de 240 pacientes sobre 10 variables de tipo numéricas y factores.

En primer lugar debemos hacer una selección del dataset con la información de 150 individuos escogidos de forma aleatoria del dataset original `armd.wide`, a partir de las dos últimas cifras del NIF, en este caso *41*, siguiendo las siguientes indicaciones y lo guardamos en el objeto `my.armd.wide`:

```
# fijamos la semilla para el generador pseudoaleatorio
set.seed(params$seed.train)
# creamos la muestra de 150 individuos
muestra<-sample(1:nrow(armd.wide),params$sp.train)
```

```
my.armd.wide <- data.frame(armd.wide[muestra,])
dim(my.armd.wide) # vemos que se ha creado adecuadamente

[1] 150 10

write.csv(my.armd.wide, "./my.armd.wide.csv") # guardamos la selección
head(my.armd.wide)
```

	subject	lesion	line0	visual0	visual4	visual12	visual24	visual52	treat.f	miss.pat
168	168	2	11	52	47	45	36	20	Active	—
163	163	2	12	58	43	NA	NA	NA	Active	-XXX
5	5	1	14	70	NA	NA	NA	NA	Active	XXXX
96	96	1	17	84	83	83	85	84	Placebo	—
126	126	3	9	47	52	58	51	52	Active	—
72	72	3	8	41	51	46	41	26	Placebo	—

Con los datos de estos pacientes, reducimos los otros dos conjuntos de datos de `armd` y `armd0`, para que contengan los datos de los mismos pacientes de mi selección. Tenemos por tanto, tres conjuntos de datos de 150 pacientes en tres diferentes data.frame:

```
library(dplyr)
my.armd0 <- filter(armd0, subject %in% my.armd.wide$subject)
#write.csv(my.armd0, "./my.armd0.csv") # guardamos la selección
dim(my.armd0)
```

```
[1] 684 8

head(my.armd0)
```

subject	treat.f	visual0	miss.pat	time.f	time	visual	tp
1	Active	59	-XX	Baseline	0	59	0
1	Active	59	-XX	4wks	4	55	1
1	Active	59	-XX	12wks	12	45	2
2	Active	65	—	Baseline	0	65	0
2	Active	65	—	4wks	4	70	1
2	Active	65	—	12wks	12	65	2

```
library(dplyr)
my.armd <- filter(armd, subject %in% my.armd.wide$subject)
#write.csv(my.armd, "./my.armd.csv") # guardamos la selección
dim(my.armd)
```

```
[1] 534 8

head(my.armd)
```

subject	treat.f	visual0	miss.pat	time.f	time	visual	tp
1	Active	59	-XX	4wks	4	55	1
1	Active	59	-XX	12wks	12	45	2
2	Active	65	—	4wks	4	70	1
2	Active	65	—	12wks	12	65	2
2	Active	65	—	24wks	24	65	3
2	Active	65	—	52wks	52	55	4

Como vemos en estos dos últimos dataset cada sujeto tiene varias filas de datos según el número de medidas que se hayan realizado a lo largo del estudio.

Podemos determinar cuáles de las características actúan como factores con el siguiente script:

```
(facs <- sapply(my.armd.wide, is.factor))

subject  lesion    line0  visual0  visual4  visual12  visual24  visual52  treat.f  miss.pat
  TRUE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    FALSE    TRUE    TRUE

names(facs[facs == TRUE])

[1] "subject" "treat.f" "miss.pat"
```

Especialmente interesante es el factor `miss.pat`, creado con los valores de las columnas `visual4`, `visual12`, `visual24`, y `visual52`. Los elementos del vector `miss.pat` resultante indican que, por ejemplo, para el paciente 163 en el marco de datos faltaban mediciones de agudeza visual en la semana 12, 24 y 52, mientras que para paciente 5, no se obtuvieron mediciones de agudeza visual en cualquier visita posterior a la aleatorización inicial.

2.2. Exploración de la muestra estudio

2.2.1. ARMD Trial: Agudeza visual

En los datos ARMD, estamos interesados principalmente en el efecto del tratamiento sobre las mediciones de agudeza visual. Por lo tanto, en la siguiente figura, primero observamos las mediciones visuales (eje y) trazándolas respecto al tiempo (eje x) para varios pacientes seleccionados de ambos grupos de tratamiento, placebo y activo. Más específicamente, se seleccionan 10 pacientes de cada grupo. En base a los gráficos que se muestran en la figura 1, se pueden realizar varias observaciones:

```
library(lattice)
my.armd0.subset <- # Subset
  subset(my.armd0, as.numeric(subject) %in% seq(1, 150, 10))
xy1 <- # Draft plot
  xyplot(visual ~ jitter(time) | treat.f,
    groups = subject,
    data = my.armd0.subset,
    type = "l", lty = 1)
update(xy1,
  xlab = "Time (in weeks)",
  ylab = "Visual acuity",
  grid = "h")

library(ggplot2) # Alternativa
ggplot(my.armd0.subset, aes(x=time, y=visual, group=subject, color=treat.f)) +
  geom_line()+geom_point()+xlab("Tiempo (semanas)") + ylab("Agudeza Visual")
```

- En general, la agudeza visual tiende a disminuir con el tiempo, es decir, los pacientes con ARMD pierden progresivamente la visión.
- pero hay perfiles individuales que se desvían fuertemente de una tendencia lineal.
- La agudeza visual al inicio parece determinar, al menos parcialmente, el nivel general de las mediciones posteriores a la aleatorización.
- Hay pacientes con datos faltantes.

2.2.2. Patrones de datos faltantes

En este paso inspeccionamos los patrones de datos faltantes en los datos de `my.armd.wide` para las mediciones de agudeza visual posteriores a la aleatorización utilizando tres métodos diferentes

Como hemos visto en el apartado anterior, la variable factor `miss.pat` indica cuál de las cuatro medidas posteriores a la aleatorización faltan para un paciente en particular. Podemos emplear tres métodos diferentes para tabular el número de pacientes con diferentes niveles del factor `miss.pat`:

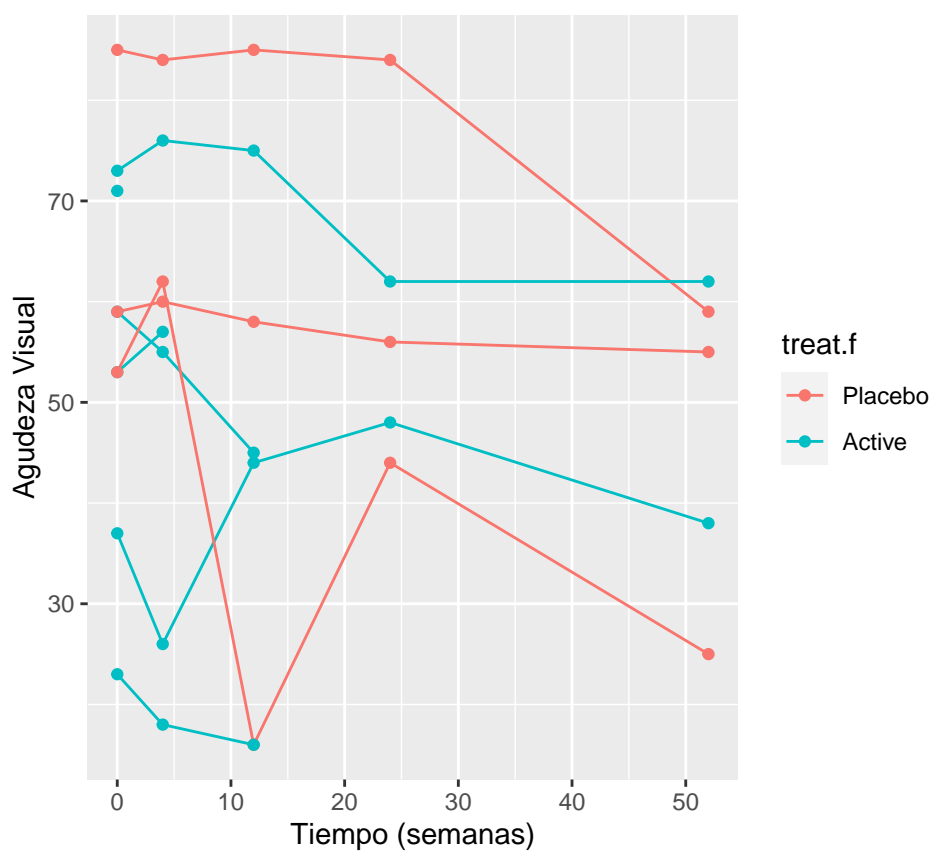
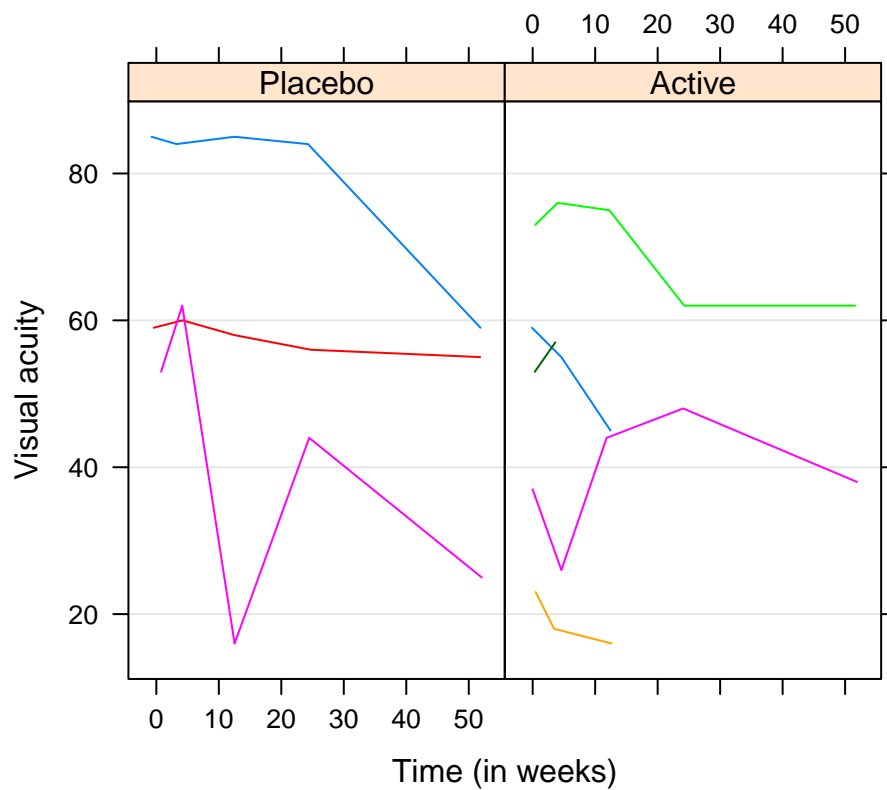


Figura 1: Perfiles de agudeza visual para pacientes seleccionados (spaghetti plot")

```
table(my.armd.wide$miss.pat)
```

—	—X	-X-	-XX	-XX-	-XXX	X—	X-XX	XXXX
118	11	3	5	1	4	1	1	6

```
with(my.armd.wide, table(miss.pat))
```

—	—X	-X-	-XX	-XX-	-XXX	X—	X-XX	XXXX
118	11	3	5	1	4	1	1	6

```
xtabs(~miss.pat, my.armd.wide)
```

—	—X	-X-	-XX	-XX-	-XXX	X—	X-XX	XXXX
118	11	3	5	1	4	1	1	6

De los resultados mostrados, podemos concluir que, en nuestra muestra, hubo 118 pacientes con las cuatro mediciones de agudeza visual posteriores a la aleatorización y 6 pacientes para quienes faltaban las cuatro mediciones. Además hay seis ($= 3 + 1 + 1 + 1$) pacientes con cuatro patrones diferentes de datos faltantes no monótonos, es decir, con mediciones intermitentes de agudeza visual faltante, lo que será importante tener en cuenta al crear el modelo con estos patrones.

2.2.3. Perfiles de valor medio

Vemos en primer lugar los datos que están completos (sin NA) y posteriormente calculamos las medias de la muestra de las mediciones de agudeza visual para diferentes visiones y grupos de tratamiento.

```
# Recuentos de mediciones de agudeza visual que no faltan
```

```
attach(my.armd0)
```

```
flst <- list(time.f, treat.f) # por factores
```

```
(tN <- # conteos
```

```
tapply(visual, flst,
```

```
FUN = function(x) length(x[!is.na(x)])))
```

	Placebo	Active
Baseline	68	82
4wks	66	76
12wks	66	73
24wks	63	67
52wks	60	63

Obtenemos una matriz con el número de mediciones de agudeza visual que no faltan para cada visita y cada grupo de tratamiento (**tN**). Como vemos la matriz indica que no faltaron mediciones al inicio y en general, hay más datos faltantes en el grupo del tratamiento activo.

En la siguiente tabla calculamos las medias (**tMn**) y las medianas (**tMd**) de la muestra de las mediciones de agudeza visual para grupo de tratamiento (**treat.f**).

La matriz resultado (**res**) muestra que, en promedio, hay muy poca diferencia en la agudeza visual entre los dos grupos de tratamiento al inicio del estudio. Durante el curso del estudio, la agudeza visual media disminuyó con el tiempo en ambos grupos y el valor medio mayor en el grupo placebo al final del estudio, pero menor al inicio.

```
# Muestra de medias y medianas de mediciones de agudeza visual
attach(my.armd0)
tMn <- tapply(vvisual, flst, FUN = mean) # media de las muestras
tMd <- tapply(vvisual, flst, FUN = median) # medianas de las muestras
colnames(res <- cbind(tN, tMn, tMd)) # nombre de las columnas

[1] "Placebo" "Active" "Placebo" "Active" "Placebo" "Active"

nms1 <- rep(c("P", "A"), 3)
nms2 <- rep(c("n", "Mean", "Media"), rep(2, 3))
colnames(res) <- paste(nms1, nms2, sep = ":") # nombre de las nuevas columnas
res
```

	P:n	A:n	P:Mean	A:Mean	P:Media	A:Media
Baseline	68	82	53.20588	56.04878	54.5	58.5
4wks	66	76	51.68182	52.44737	53.0	54.5
12wks	66	73	49.63636	50.50685	51.0	52.0
24wks	63	67	47.39683	47.41791	46.0	48.0
52wks	60	63	45.00000	40.55556	46.5	38.0

La siguiente figura muestra gráficos de caja para la agudeza visual de los cinco puntos temporales y los dos grupos de tratamiento.

```
library(lattice)
bw1 <- # plot
bwplot(vvisual ~ time.f | treat.f,
data = my.armd0)
xlims <- c("Base", "4\nwks", "12\nwks", "24\nwks", "52\nwks")
update(bw1, xlim = xlims, pch = "|") # Plot final
```

El gráfico anterior ilustra los patrones implicados por las medias y medianas de muestra. La disminución de los valores medios en el tiempo se ve claramente para ambos grupos de tratamiento, aunque es más pronunciado para el grupo de tratamiento activo. Ya que encontramos más datos faltantes en ese grupo, una posible explicación podría ser que los pacientes cuya agudeza visual mejoró abandonaron el estudio, lo que llevaría a una progresión más rápida de la enfermedad en ese grupo de tratamiento.

Para comprobar esa posibilidad, se crea la siguiente figura que muestra los valores medios de agudeza visual para pacientes con diferentes patrones de datos faltantes, junto con el número de sujetos para cada patrón. Observamos que el número de sujetos con patrones de un mayor número de valores faltantes tiende a ser menor.

```
mnt.pat<- # patrones monótonos
c("----", "---X", "--XX", "-XXX", "XXXX")
armd.mnt <- # Data subset
subset(my.armd, miss.pat %in% mnt.pat)

## Eliminar niveles no utilizados del factor miss.pat
armd.mnt1 <-
within(armd.mnt,
{
miss.pat <- factor(miss.pat, levels=mnt.pat)
})

library(lattice)
my.armd.subset1 <- # Subset
subset(armd.mnt1, as.numeric(subject) %in% seq(1, 240, 10))
xy1 <- # plot
xyplot(vvisual ~ jitter(time) | treat.f,
```

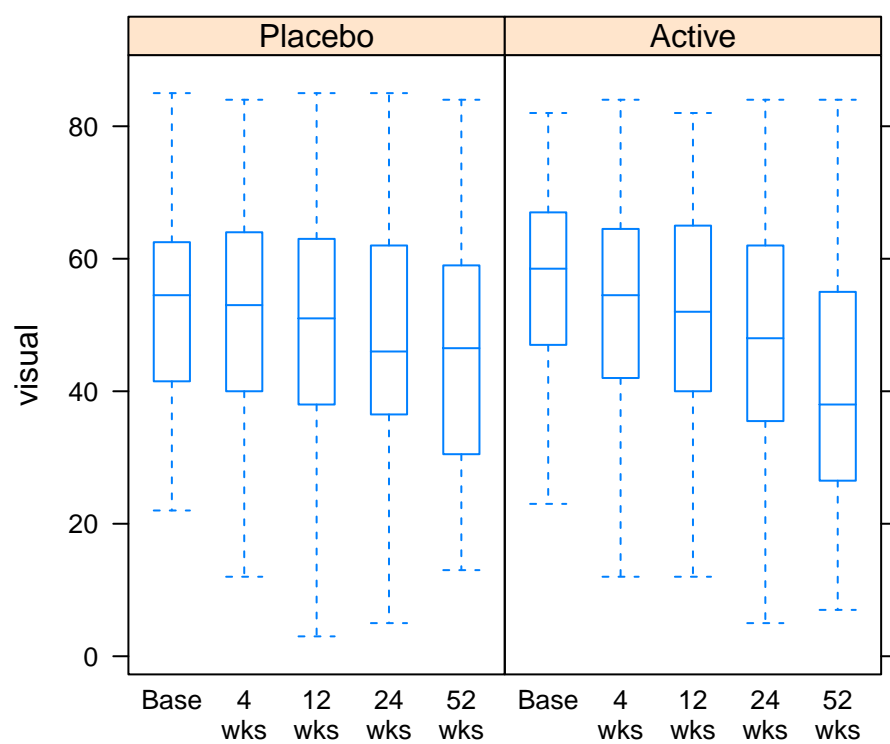


Figura 2: Boxplots de la agudeza visual por tratamiento y tiempo


```

groups = subject,
data = my.armd.subset1,
type = "l", lty = 1,
panel=function(x, y, ...) {
  panel.xyplot(x, y, ...);
  ltext(x=x, y=y, labels=my.armd.subset1$subject, pos=1, offset=2, cex=0.6)
})
update(xy1,
  xlab = "Time (in weeks)",
  ylab = "Visual acuity",
  grid = "h")

```

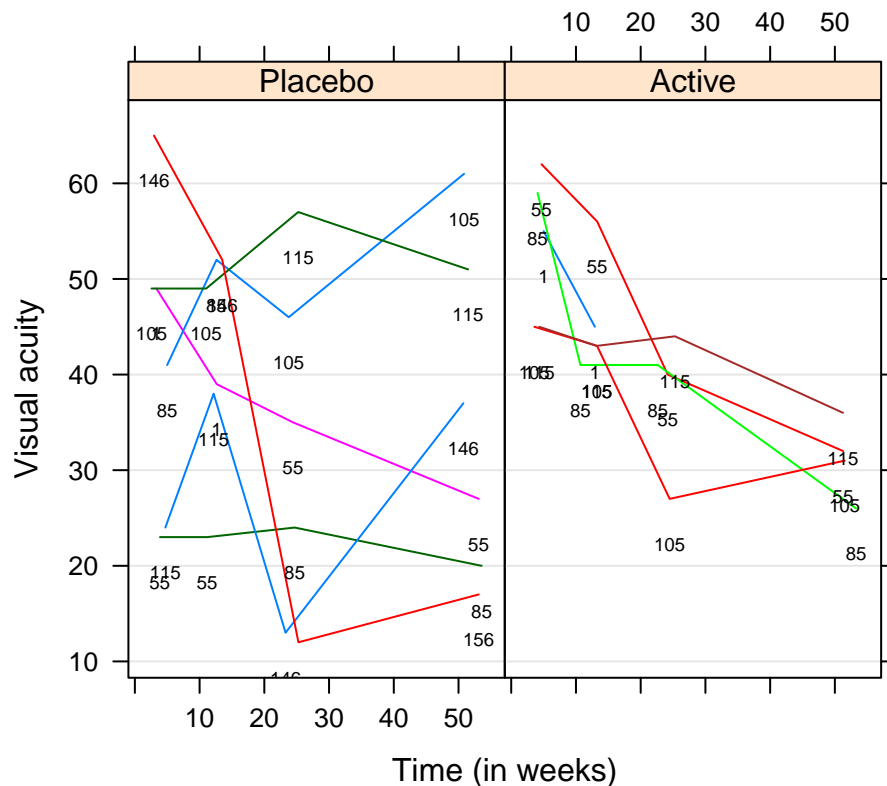


Figura 3: Perfiles medios de agudeza visual por patrón y tratamiento (solo patrones de datos faltantes monótonos)

Como se observa en la figura anterior (Figura 3) los perfiles medios, disminuyen para la mayoría de los patrones, pero en general, no sugieren una mejora en la agudeza visual antes del descenso. Por lo tanto, parece que no se respalda la hipótesis anterior de una disminución más rápida de la agudeza visual media en el grupo de tratamiento activo.

En el paso siguiente buscamos el número y forma de los patrones de datos faltantes monótonos para la agudeza visual. Para ello empleamos el siguiente script que crea una matriz que contiene el número de pacientes para cada patrón de datos faltantes monótonos y para cada grupo de tratamiento. Vemos que hay 144 pacientes de 150 que contienen patrones monótonos.

Los resultados mostrados indican que los perfiles de valor medio para patrones de datos faltantes con un mayor número de valores faltantes, se basan en mediciones para un pequeño número de pacientes. Por lo tanto, la variabilidad de estos perfiles es mayor que para los patrones con un menor número de valores faltantes.

```

## El número de pacientes por tratamiento y patrón de datos faltantes (solo patrones monótonos)
## Subset of the data with monotone missing-data patterns
mnt.pat<- # Monotone patterns
c("----", "---X", "--XX", "-XXX", "XXXX")
armd.wide.mnt <- # Data subset
subset(my.armd.wide, miss.pat %in% mnt.pat)
dim(armd.wide.mnt) # Number of rows and cols

[1] 144 10

levels(armd.wide.mnt$miss.pat) # Some levels not needed

[1] "----" "---X" "--XX" "-XXX" "-XX-" "-XXX" "X---" "X-XX" "XXXX"

## Removing unused levels from the miss.pat factor
armd.wide.mnt1 <-
within(armd.wide.mnt,
{
miss.pat <- factor(miss.pat, levels=mnt.pat)
})
levels(armd.wide.mnt1$miss.pat)

[1] "----" "---X" "--XX" "-XXX" "XXXX"

## The number of patients with different monotone missing-data patterns
with(armd.wide.mnt1,
{
fl <- list(treat.f, miss.pat) # List of "by" factors
tapply(subject, fl, FUN=function(x) length(x[!is.na(x)]))
})

```

	—	—X	—XX	—XXX	XXXX
Placebo	57	5	1	1	1
Active	61	6	4	3	5

2.2.4. Varianzas y correlaciones de las mediciones de agudeza visual

La siguiente figura (fig. 4) muestra una matriz de diagrama de dispersión para las mediciones de agudeza visual de los pacientes que no poeen datos faltantes (n=118). Los diagramas de dispersión para los pares de variables correspondientes se muestran debajo de la diagonal y el tamaño de la fuente para los coeficientes de correlación que se muestran por encima de la diagonal es proporcional a su valor.

```

my.lowerPanel <- ## pairwise.complete.obs
function(x, y, subscripts, ...){
panel.grid(h = -1, v = -1)
panel.xyplot(x, y, ...)
}
my.upperPanel <- ## pairwise.complete.obs
function(x, y, subscripts, ...){
panel.xyplot(x, y, type = "n", ...)
corx <- round(cor(x,y,use = "complete.obs"),2)
abs.corx <- abs(corx)
cex.value <- 3
ltext(50,50, corx, cex = abs.corx* cex.value)
}
mySuperPanel <- function(z, subscripts, panel.subscripts,...){
# z is data frame. Abbreviated variable names on the diagonal of the splom.
panel.pairs(z, subscripts = subscripts,

```

```

panel.subscripts = panel.subscripts,
as.matriz = TRUE,
upper.panel = "my.upperPanel",
lower.panel = "my.lowerPanel",
prepanel.limits = function(z) return(c(1, 90))
})

splom.form <- formula( ~cbind(vis0, vis4, vis12, vis24, vis52))
armd.wide.tmp <- subset(my.armd.wide, miss.pat == "----",
select = c(visual0, visual4, visual12, visual24, visual52))
names(armd.wide.tmp) <- c("vis0", "vis4", "vis12", "vis24", "vis52") # Short names

splom.object <- splom(splom.form,
data = armd.wide.tmp,
par.settings = list(fontsize = list(points = 4), axis.text = list(cex = 0.9)),
as.matriz = TRUE,
xlab = "",
superpanel = mySuperPanel
)
print(splom.object)

```

Se puede observar que las mediciones adyacentes en el tiempo están fuertemente correlacionadas, con valores de correlación elevados. Dicha correlación disminuye cuando el intervalo de tiempo aumenta. Existe una correlación positiva significativa entre la agudeza visual al inicio y en las otras mediciones posteriores. Por lo tanto, los valores de referencia podrían usarse para explicar la variabilidad general de las observaciones posteriores a la aleatorización.

Se calculan ahora las estimaciones de las matrices de varianza-covarianza y correlación para las mediciones de agudeza visual y mostramos los valores de la diagonal:

```

visual.x <- subset(my.armd.wide, select = c(visual0:visual52)) # selección de datos visuales
(varx <- var(visual.x, use = "complete.obs")) # Var-cov matriz

```

	visual0	visual4	visual12	visual24	visual52
visual0	244.4732	229.2570	232.8866	221.2269	177.2176
visual4	229.2570	267.9655	255.9477	239.6558	199.3527
visual12	232.8866	255.9477	330.2785	281.7697	242.3585
visual24	221.2269	239.6558	281.7697	344.5456	293.4801
visual52	177.2176	199.3527	242.3585	293.4801	340.2037

```

# use = "complete.obs" selecciona solo aquellas filas del marco de datos `visual.x`
# que no contienen ningún valor faltante
print(cor(visual.x, use = "complete.obs"), # matriz de Correlación
digits = 2)

```

	visual0	visual4	visual12	visual24	visual52
visual0	1.00	0.90	0.82	0.76	0.61
visual4	0.90	1.00	0.86	0.79	0.66
visual12	0.82	0.86	1.00	0.84	0.72
visual24	0.76	0.79	0.84	1.00	0.86
visual52	0.61	0.66	0.72	0.86	1.00

```
diag(varx) # Var-cov diagonal elements
```

```

visual0 visual4 visual12 visual24 visual52
244.4732 267.9655 330.2785 344.5456 340.2037

```

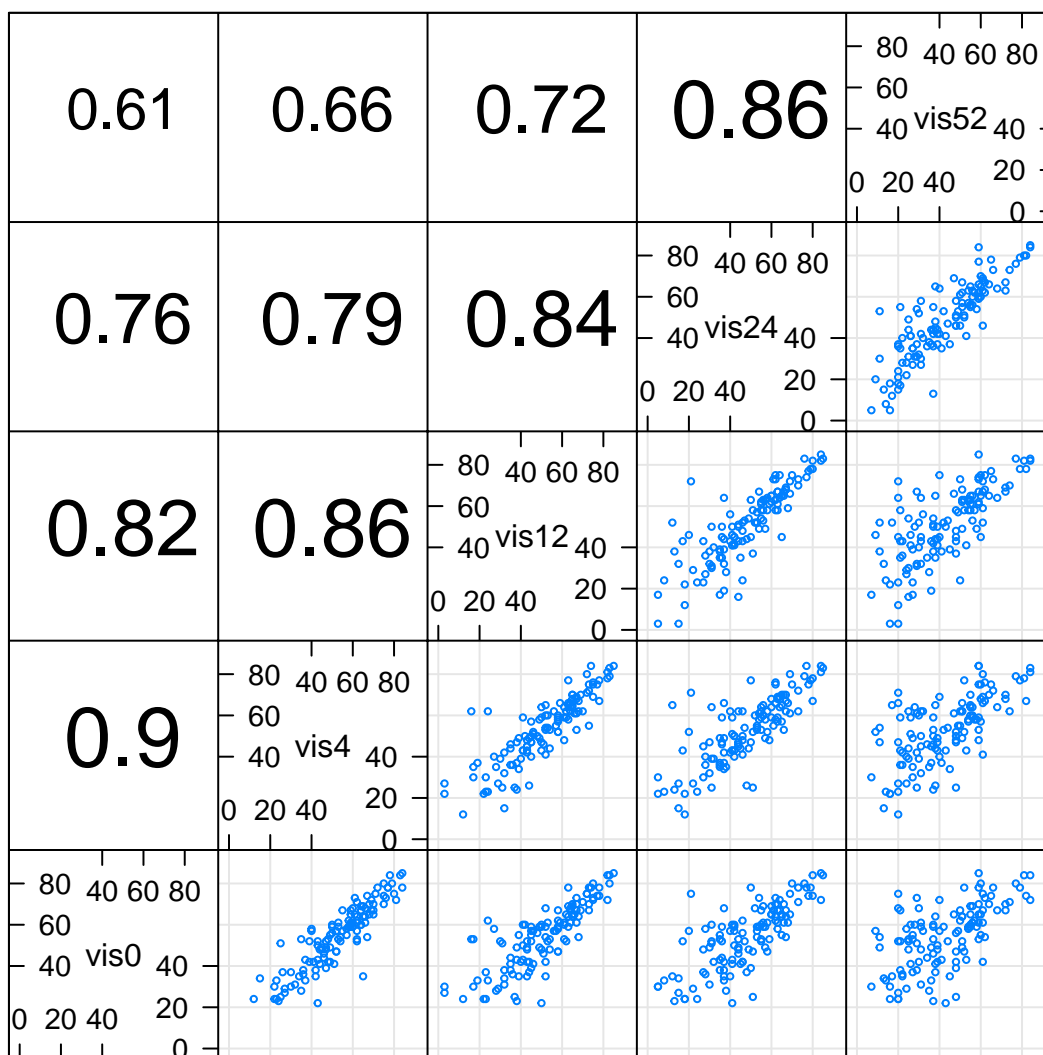


Figura 4: Matriz de diagrama de dispersión para mediciones de agudeza visual, con los coeficientes de correlación (sobre la diagonal) para casos completos ($n = 118$)

```
# cov2cor(varx) # Corr mtx (alternative way)

# alternativa de calculo de la matriz: los elementos de las matrices se estimarían
# utilizando datos de todos los pacientes con observaciones completas para el
# par particular de mediciones de agudeza visual. Esto podría dar como resultado
# estimaciones de var-cov o m. corr, que podrían no ser semidefinidas positivas

# (varx2 <- var(visual.x, use = "pairwise.complete.obs"))
# print(cor(visual.x, use = "pairwise.complete.obs"), # Corr mtx
# digits = 2)
# diag(varx2)
```

La matriz de correlación estimada sugiere una correlación significativa entre las mediciones. También observamos que la correlación disminuye claramente con el intervalo de tiempo, como se ha mostrado también en puntos anteriores.

3. Modelos lineales mixtos (LMM)

Vamos a aplicar un modelo a nuestros datos que permita tener en cuenta la correlación entre las mediciones, utilizando modelos lineales de efectos mixtos (LMM). Un modelo simple asume que la relación entre las variables es la misma en todos los sujetos. Sin embargo, normalmente dicho modelo no es el mismo en todos ellos, por ello en este enfoque se aplican efectos aleatorios que describen la contribución de la variabilidad en diferentes niveles de las observaciones de forma que se predice la relación entre las variables en un sentido general a la población, sin limitarnos a los datos concretos del estudio.

Ajustamos varios LMM a la selección de datos de ARMD de nuestro estudio. Para ello vamos a emplear principalmente la función `lme()` del paquete `nlme` y por último algunos ajustados con la función `lmer()` del paquete `lme4.0`.

3.1. Modelo con intersecciones aleatorias y varianza residual homogénea

Para elegir la estructura de los efectos aleatorios es necesario incluir todos los posibles términos fijos y sus interacciones en el modelo. Luego se comparan distintos modelos que varían en sus efectos aleatorios pero que mantienen la misma estructura fija por medio de REML. Sin embargo, en este caso vamos a aplicar directamente el modelo propuesto por Gałecki and Burzykowski (2013).

En este modelo, asumimos que sólo hay una línea de regresión con una única constante y una única pendiente. La constante α y la pendiente β son los parámetros fijos del modelo. Además, hay una constante aleatoria, a_j , que añade cierta cantidad de variación a la constante del modelo general en cada una de los sujetos. Se asume que esta constante aleatoria sigue una distribución normal de media 0 y varianza σ_a^2 . Por lo tanto, los parámetros estimados en el modelo son cuatro, α , β , la varianza residual σ^2 y la varianza de la constante σ_a^2 . De esta forma tenemos un modelo equivalente al modelo lineal, pero en donde sólo es necesario estimar cuatro parámetros, en vez del mismo número de parámetros que el número de sujetos estudiados. Lo que estimamos en este modelo es la varianza de esta distribución. Empleamos para ello la función `lme()` tal y como se realiza en la página 331 del capítulo 16 del libro Gałecki and Burzykowski (2013) pero para nuestro conjunto de datos.

La fórmula `lmm1` que define el modelo incluye 4 variables, incluida una interacción entre el tiempo y el tipo de tratamiento (`time:treat.f`). El factor `treat.f` se parametriza con “Activo” como nivel de referencia. Por defecto, `lme()` asume errores residuales independientes con una varianza constante σ^2 . Podemos ver la tabla de efectos fijos estimada utilizando el Función `printCoefmat()`. El modelo por defecto se crea según Estimación por Máxima Verosimilitud Restringida (REML):

```
library(nlme)
lmm1.form <- # formula general del modelo, parte fija
formula(visual ~ visual0 + time + treat.f + treat.f:time)
(lmm1 <- # se aplica el modelo,
lme(lmm1.form,
random = ~ 1|subject, data = my.armd)) # al grupo de datos reducido my.armd
```

```
Linear mixed-effects model fit by REML
Data: my.armd
Log-restricted-likelihood: -2004.824
Fixed: lmm1.form
      (Intercept)          visual0          time          treat.fActive
      5.1239173          0.8781773          -0.1502096          -0.6318769
time:treat.fActive
      -0.1206905
```

```
Random effects:
Formula: ~1 | subject
      (Intercept) Residual
StdDev:    8.915912 8.218683
```

```
Number of Observations: 534
Number of Groups: 144
```

```
# argumento `random = ~ 1 | subject` especifica intercepciones aleatorias
# específicas del sujeto
summary(lmm1)
```

```
Linear mixed-effects model fit by REML
Data: my.armd
      AIC      BIC    logLik
4023.648 4053.545 -2004.824
```

```
Random effects:
Formula: ~1 | subject
      (Intercept) Residual
StdDev:    8.915912 8.218683
```

```
Fixed effects: list(lmm1.form)
      Value Std.Error DF t-value p-value
(Intercept)    5.123917  3.186086 388  1.608217  0.1086
visual0         0.878177  0.054308 141 16.170421  0.0000
time            -0.150210  0.028951 388 -5.188462  0.0000
treat.fActive   -0.631877  1.879734 141 -0.336152  0.7373
time:treat.fActive -0.120690  0.040310 388 -2.994068  0.0029
Correlation:
      (Intr) visul0 time   trt.fA
visual0      -0.903
time         -0.197 -0.003
treat.fActive -0.233 -0.087  0.339
time:treat.fActive 0.141  0.003 -0.718 -0.462
```

```
Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-3.48260927 -0.35151744  0.03257478  0.50366666  3.20774570
```

```
Number of Observations: 534
Number of Groups: 144
```

```
printCoefmat(summary(lmm1)$tTable, # estimated fixed-effects
has.Pvalue = TRUE, P.values = TRUE) # ... with p-values
```

```
      Value Std.Error DF t-value p-value
(Intercept)    5.123917  3.186086 388.000000  1.6082  0.108601
visual0         0.878177  0.054308 141.000000 16.1704 < 2.2e-16 ***
```

```

time                -0.150210    0.028951 388.000000 -5.1885 3.425e-07 ***
treat.fActive        -0.631877    1.879734 141.000000 -0.3362 0.737256
time:treat.fActive   -0.120690    0.040310 388.000000 -2.9941 0.002929 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Los resultados anteriores muestran que la varianza residual es $\sigma^2 = 8.2187$ y la varianza de la constante $\sigma_a^2 = 8.9159$. Para la parte de los efectos fijos del modelo, α (constante) + $\beta \cdot$ variables, la constante se estima en $\alpha = 5.1239$ y la pendiente en β de $\text{visual0} = 0.8782$. Ambos parámetros son significativamente distintos de 0. La correlación entre la constante y la pendiente de cada variable es pequeña.

Para cada sujeto, la constante aumenta o disminuye por un valor aleatorio. Este valor aleatorio sigue una distribución normal con media esperada 0 y varianza $\sigma_a^2 = 8.9159$. La varianza residual, es decir, el error que podemos añadir a cada una de nuestras predicciones, viene dada por $\sigma^2 = 8.2187$.

Los valores p, que corresponden a las estadísticas de la prueba t para los coeficientes de efectos fijos (fixed-effects coefficients), los vemos también en la salida anterior, con valores significativos para *Visual0*, *time* y la interacción, sin embargo no es significativo para el tipo de tratamiento (placebo y activo).

En el siguiente script, con la función `getGroupsFormula()` obtenemos información sobre la jerarquía de datos implicada por el modelo ajustado. Indica un único nivel de agrupación, definido por los niveles del sujeto. Como vemos el factor de agrupación tiene 144 niveles (sujetos). Por ejemplo, del primer sujeto tenemos dos observaciones, para el segundo y el tercero cuatro observaciones, etc. Además obtenemos el mínimo de observaciones, que sería 1 y el máximo de observaciones es 4, como ya sabemos.

```

getGroupsFormula(lmm1) # formula

~subject
<environment: 0x0000000025f10680>

str(grpF <- getGroups(lmm1)) # agrupamos por factor

Factor w/ 144 levels "1","2","4","6",...: 1 1 2 2 2 2 3 3 3 3 ...
- attr(*, "label")= chr "subject"

grpF[1:17]

[1] 1 1 2 2 2 2 4 4 4 4 6 6 6 6 9 9 9
144 Levels: 1 2 4 6 9 10 15 16 17 18 19 20 22 24 26 27 30 31 38 39 40 43 45 49 50 ... 240

levels(grpF)[1:5]

[1] "1" "2" "4" "6" "9"

range(xtabs(~grpF))

[1] 1 4

```

Para obtener más información sobre la estructura estimada de varianza-covarianza para efectos aleatorios(D) y errores residuales (Ri) del modelo, utilizamos las funciones `getVarCov()` y `VarCorr()`. El argumento `individual = "2"`, utilizado en función `getVarCov()`, solicita la matriz de varianza-covarianza de efectos aleatorios para el segundo individuo, es decir, sujeto == 2, en el conjunto de datos analizado. Sin embargo, en nuestro caso, el número de sujeto no es importante, ya que se supone que la estructura de varianza-covarianza de los efectos aleatorios es la misma para todos los individuos.

```

## la matriz D estimada
getVarCov(lmm1, individual = "2") # valor de la varianza únicamente

```

```

Random effects variance covariance matrix
(Intercept)
(Intercept)      79.493
Standard Deviations: 8.9159

```

```
VarCorr(lmm1)
```

```
subject = pdLogChol(1)
      Variance StdDev
(Intercept) 79.49349 8.915912
Residual    67.54676 8.218683
```

```
## La matriz Ri estimada
```

```
getVarCov(lmm1,
type = "conditional", # matriz estimada de varianza-covarianza Ri de los
# errores aleatorios residuales
individual = "2")
```

```
subject 2
Conditional variance covariance matrix
      1      2      3      4
1 67.547 0.000 0.000 0.000
2 0.000 67.547 0.000 0.000
3 0.000 0.000 67.547 0.000
4 0.000 0.000 0.000 67.547
Standard Deviations: 8.2187 8.2187 8.2187 8.2187
```

Como vemos este sujeto (número 2) tiene las cuatro mediciones de agudeza visual posteriores a la aleatorización, por lo que nos proporciona una matriz de 4×4 . Debido a que el modelo creado supone errores residuales independientes con la misma varianza en todos los tiempos de medición, se muestra una matriz diagonal $Ri = \sigma^2 I_4 = 67,547 \times I_4$

En el siguiente script calculamos la matriz de varianza-covarianza marginal estimada, aplicando la función `getVarCov()` con el argumento `type = "marginal"`. La varianza marginal se estima mediante la suma de la varianza residual estimada $\sigma^2 = 67,547$ y la varianza de las intersecciones aleatorias $d = 79.493$.

La matriz de correlación marginal estimada implica un coeficiente de correlación positivo constante de 0.540624 para cualquiera de las dos mediciones de agudeza visual obtenidas para el mismo paciente en diferentes puntos de tiempo:

```
# La matriz de varianza-covarianza marginal estimada y la matriz de correlación
# correspondiente al modelo 1
```

```
(lmm1.cov <-
getVarCov(lmm1,
type = "marginal", # Vi
individual = "2"))
```

```
subject 2
Marginal variance covariance matrix
      1      2      3      4
1 147.040 79.493 79.493 79.493
2 79.493 147.040 79.493 79.493
3 79.493 79.493 147.040 79.493
4 79.493 79.493 79.493 147.040
Standard Deviations: 12.126 12.126 12.126 12.126
```

```
(cov2cor(lmm1.cov[[1]])) # Corr(Vi)
```

	1	2	3	4
1	1.000000	0.540624	0.540624	0.540624
2	0.540624	1.000000	0.540624	0.540624
3	0.540624	0.540624	1.000000	0.540624
4	0.540624	0.540624	0.540624	1.000000

La comparación de modelos con las pruebas de razón de probabilidad es una mejor manera de probar si un parámetro

es significativo. Es decir, si agregar el parámetro a su modelo mejora significativamente el ajuste del modelo y si ese parámetro debe incluirse en el modelo. Vamos a estimar otros modelos y a compararlos con este primer modelo y probar su eficacia.

3.2. Modelo con intercepciones aleatorias y la función `varPower()`, de varianza residual

Con el modelo anterior creamos este nuevo modelo `lmm2` utilizando el argumento `weights = varPower (form = ~ time)`. Para especificar el nuevo modelo, utilizamos la misma parte de efectos fijos que en el modelo `lmm1`, pero modificamos la estructura de varianza-covarianza de los errores aleatorios residuales con el uso de la función de varianza `varPower()`. La matriz diagonal `Ri` proporcionada por `getVarCov` ahora posee elementos desiguales definidos por la función `varPower()`. Debido a que la varianza cambia con el tiempo, los coeficientes de correlación marginal entre las observaciones realizadas en diferentes momentos ya no son iguales.

Veamos los parámetros del modelo:

```
# Model 2 fitted using the function lme().
```

```
(lmm2 <-  
update(lmm1,  
weights = varPower(form = ~ time),  
data = my.armd))
```

Linear mixed-effects model fit by REML

Data: my.armd

Log-restricted-likelihood: -1993.846

Fixed: lmm1.form

	(Intercept)	visual0	time	treat.fActive
	3.9327540	0.9067644	-0.1637442	-1.0738933
time:treat.fActive				-0.1054049

Random effects:

Formula: ~1 | subject

(Intercept) Residual

StdDev: 8.107852 4.099941

Variance function:

Structure: Power of variance covariate

Formula: ~time

Parameter estimates:

power

0.2513018

Number of Observations: 534

Number of Groups: 144

```
printCoefmat(summary(lmm2)$tTable, # Print fixed-effects  
has.Pvalue = TRUE, P.values = TRUE) # ... with p-values
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	3.932754	2.887552	388.000000	1.3620	0.17400
visual0	0.906764	0.049483	141.000000	18.3249	< 2.2e-16 ***
time	-0.163744	0.032174	388.000000	-5.0893	5.611e-07 ***
treat.fActive	-1.073893	1.653273	141.000000	-0.6496	0.51704
time:treat.fActive	-0.105405	0.044792	388.000000	-2.3532	0.01911 *

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

Los resultados anteriores indican que el parámetro σ es igual a 4.0999. El coeficiente de potencia δ de la función de varianza `varPower()` es igual a 0.2513. La estimación de la desviación estándar de las intersecciones aleatorias es

igual a 8.1079.

Vemos ahora las estimaciones de las matrices de varianza-covarianza asociadas con modelo lmm2: D , R_i , y V_i .

```
# Matrices D, Ri, y Vi para el modelo 2
VarCorr(lmm2) #D
```

```
subject = pdLogChol(1)
      Variance StdDev
(Intercept) 65.73726 8.107852
Residual    16.80951 4.099941
```

```
getVarCov(lmm2, #Ri
type = "conditional",
individual = "2")
```

```
subject 2
Conditional variance covariance matrix
      1      2      3      4
1 33.741  0.000  0.000  0.00
2  0.000 58.608  0.000  0.00
3  0.000  0.000 83.034  0.00
4  0.000  0.000  0.000 122.47
Standard Deviations: 5.8087 7.6556 9.1123 11.067
```

La varianza estimada de las intersecciones aleatorias es igual a 65.737, menor que el valor de 79.4934 obtenida para el modelo 1 (lmm1), lo que es normal porque, al permitir errores aleatorios residuales heterocedásticos, una gran parte de la variabilidad total se explica por las variaciones residuales. De la matriz estimada de varianza-covarianza de los errores residuales R_i se obtiene que el primer elemento diagonal de la matriz R_i es igual a $s^2 \cdot 4^{2\delta} = 4,0999 \cdot 4^{2 \cdot 0,2513} = 33,741$.

La matriz de varianza-covarianza marginal estimada V_i se muestra en la salida siguiente. La matriz de correlación marginal estimada correspondiente indica una correlación decreciente entre las mediciones de agudeza visual realizadas en puntos temporales más distantes.

```
(lmm2.cov <- #Vi
getVarCov(lmm2,
type = "marginal",
individual = "2"))
```

```
subject 2
Marginal variance covariance matrix
      1      2      3      4
1 99.478 65.737 65.737 65.737
2 65.737 124.350 65.737 65.737
3 65.737 65.737 148.770 65.737
4 65.737 65.737 65.737 188.210
Standard Deviations: 9.9739 11.151 12.197 13.719
```

```
cov2cor(lmm2.cov[[1]]) # Corr(Vi)
```

	1	2	3	4
1	1.0000000	0.5910635	0.5403677	0.4804320
2	0.5910635	1.0000000	0.4833239	0.4297153
3	0.5403677	0.4833239	1.0000000	0.3928584
4	0.4804320	0.4297153	0.3928584	1.0000000

Para resumir presento los resultados de ambos modelos en la siguiente tabla:

	Parámetro	lmm1	lmm2
Log-REML value		-2004.824	-1993.846
<i>Fixed effects</i>			
Intercept	β_0	5.123917(3.19)	3.9328(2.89)
Visual acuity at t=0	β_1	0.878177(0.05)	0.9067(0.04)
Time (in weeks)	β_2	-0.150210(0.03)	-0.1637(0.03)
Trt(Actv vs. Plcb)	β_3	-0.631877(1.88)	-1.0739(1.65)
Tm \times Treat(Actv)	β_4	-0.120690(0.04)	-0.1054(0.04)
<i>reStruct(subject)</i>			
SD(bi0)	\sqrt{d}	8.9159	8.107852
<i>Variance function</i>			
Power (TIMEd)	δ		0.2513
Scale	σ	8.2187	4.099941

3.2.1. Gráficos de diagnóstico

Vamos a graficar la bondad del ajuste de ambos modelos.

El gráfico que desarrolla la función `plot()` (fig. 5) muestra los residuales condicionales de *Pearson* versus los valores ajustados. Pero la gráfica no es muy informativa, ya que agrupa todos los residuos juntos, a pesar de que los residuos obtenidos del mismo individuo están potencialmente correlacionados. Sin embargo, puede servir para detectar, por ejemplo, valores atípicos.

```
plot(lmm2)
```

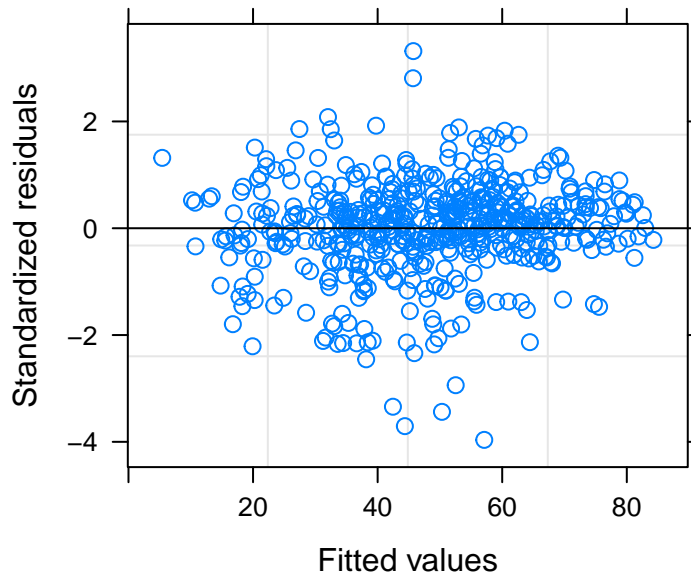


Figura 5: Gráfico residual predeterminado de residuos condicionales de Pearson

Una gráfica de los residuos para cada momento de tiempo y grupo de tratamiento podría ser más útil. Para ello se realiza el gráfico de la figura 6, donde a través del argumento `type = "pearson"` en la función `resid()` y el término `~ time | treat` se obtiene el siguiente gráfico por grupo de tratamiento a lo largo del tiempo en paneles separados, con el valor absoluto al percentil 97.5 de la distribución normal estándar por el número de la observación correspondiente a nuestros datos `my.armd`.

```
stdres.plot <-  
plot(lmm2, resid(., type = "p") ~ jitter(time) | treat.f,
```

```
id = 0.05, adj = c(-0.3, 0.5 ), grid = FALSE)
plot(update(stdres.plot,
grid = "h"))
```

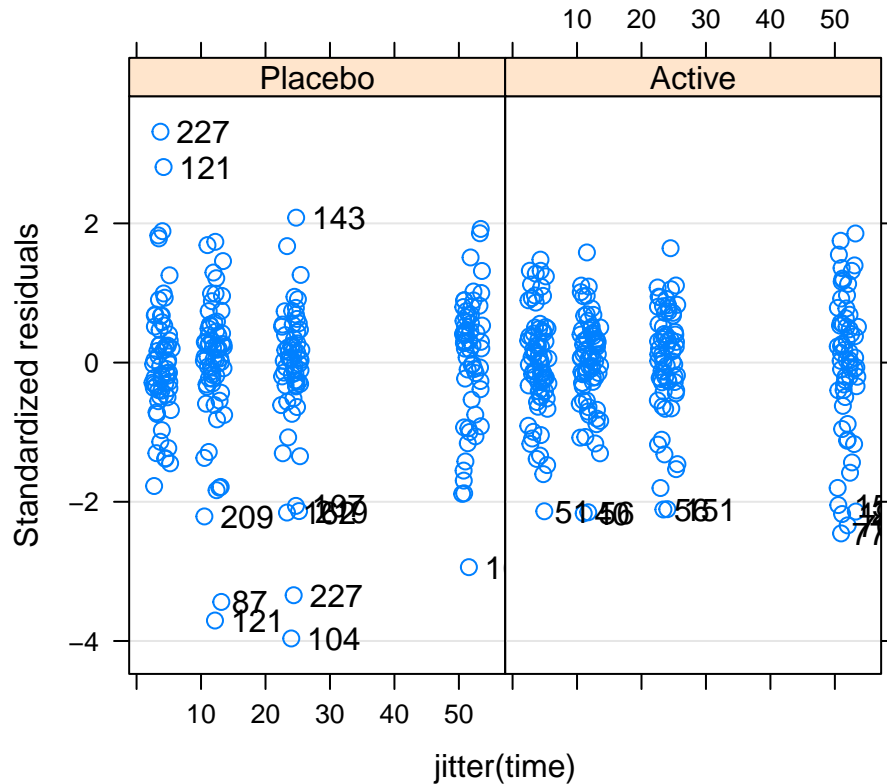


Figura 6: Gráficos de residuos de Pearson por tiempo y tratamiento

Otra opción nos la proporciona el paquete `lattice` con la función `bwplot()`:

```
library(lattice)
bw1 <- # Draft plot
bwplot(resid(lmm2, type = "p") ~ time.f | treat.f,
data = my.armd0, wend = 0.05, las = 1, id = 0.05, ylab = "Residuals", panel = function(...) {
  panel.bwplot(..., pch = "|")
  panel.xyplot(..., jitter.x = TRUE)})
xlims <- c("Base", "4\nwks", "12\nwks", "24\nwks", "52\nwks")
update(bw1, xlim = xlims, grid = FALSE)
```

Estas figuras nos permiten una evaluación de la distribución de los residuos condicionales de Pearson para cada momento del tiempo y grupo de tratamiento. A pesar de la estandarización, la variabilidad de los residuos parece variar.

El gráfico 6 también revela una serie de valores atípicos, es decir, residuos mayores, en valor absoluto, que da percentil 97.5 de la distribución normal estándar. Como vemos los valores atípicos están presentes en todos los grupos de tratamiento y en todos los puntos de tiempo.

Con el siguiente script se enumeran los sujetos para los que se marcaron los residuos periféricos en la figura anterior. Los datos incluidos en `outliers.idx` contiene las observaciones para las cuales el valor de la variable `idx` es igual a 1.

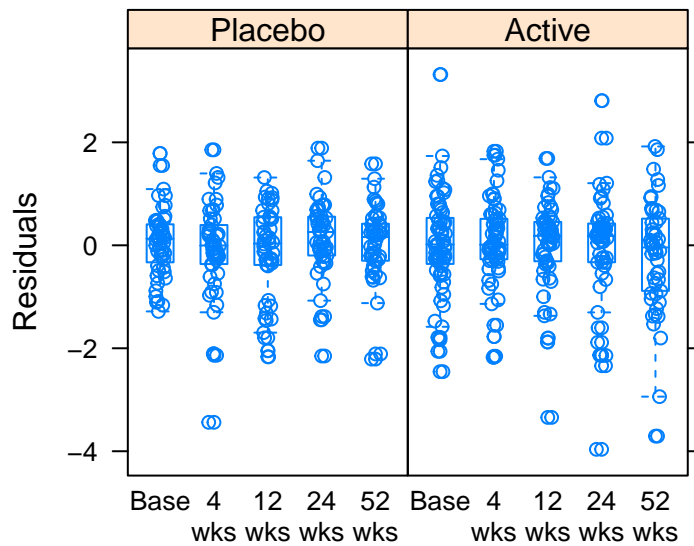


Figura 7: Boxplots de residuos de Pearson por tiempo y tratamiento

```
id <- 0.05 # Argument for qnorm()
outliers.idx <-
within(my.armd,
{
  resid.p <- resid(lmm2, type = "pearson") # Pearson resid.
  idx <- abs(resid.p) > -qnorm(id/2) # Indicator vector
})
outliers <- subset(outliers.idx, idx) # Data with outliers
nrow(outliers) # Number of outliers
```

```
[1] 22
```

```
outliers$subject # IDs of outliers
```

```
[1] 40 51 56 56 70 77 87 104 104 107 121 121 143 151 151 162 191 209 219 226 227
```

```
[22] 227
```

```
234 Levels: 1 2 3 4 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 22 23 24 25 26 27 29 ... 240
```

Hay 22 de esas observaciones y obtenemos el número de los sujetos correspondientes. Para varios sujetos, hay más de un residuo periférico, porque hay más de una medición de agudeza visual posible por sujeto.

La figura 8 muestra el gráfico Q-Q de los residuos condicionales de Pearson por momento de tiempo. Los patrones muestran algunas desviaciones de una tendencia lineal. También podemos ver el gráfico Q-Q **normal** de los efectos aleatorios, ligeramente curvilíneo, lo que podría indicar la no normalidad de los efectos aleatorios.

```
qqnorm(lmm2, ~resid(.) | time.f) # Fig. 7
qqnorm(lmm2, ~ranef(.))
```

La figura 9 muestra los valores observados y pronosticados de las mediciones de agudeza visual para pacientes seleccionados:

```
aug.Pred <-
augPred(lmm2,
primary = ~time, # Primary covariate
level = 0:1, # Marginal(0) and subj.-spec.(1)
```

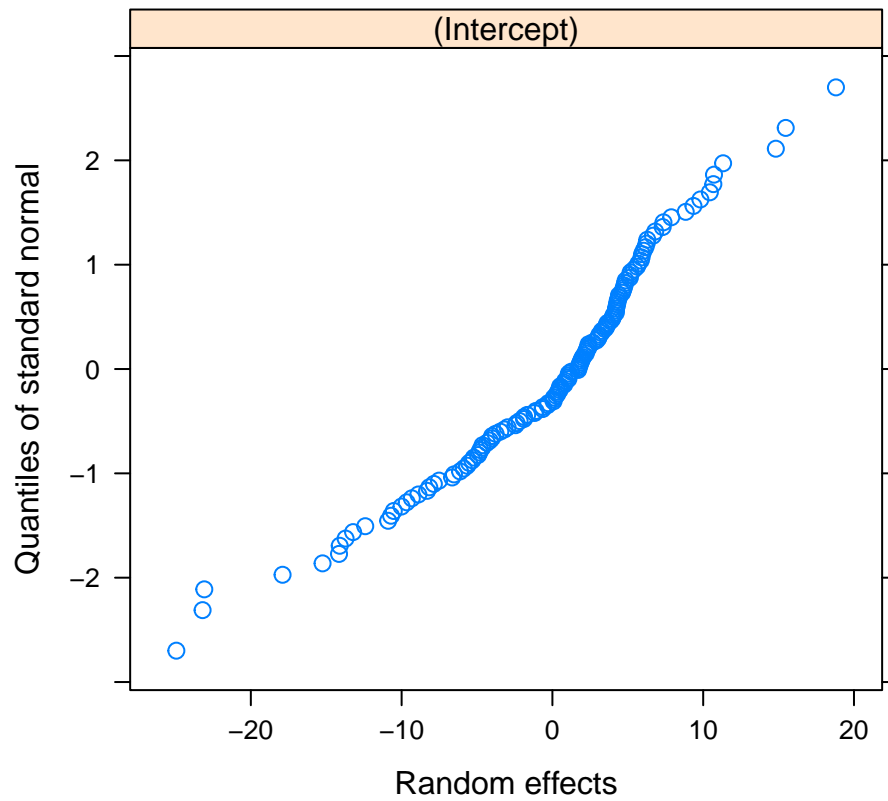
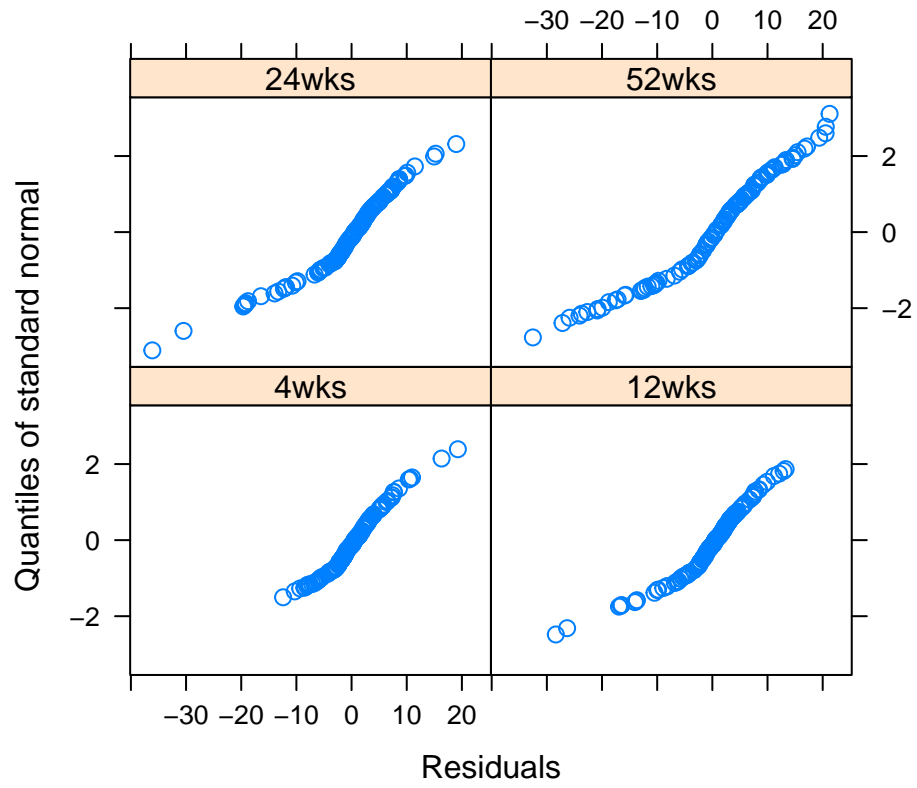


Figura 8: Gráficos Q-Q normales de residuos de Pearson y efectos aleatorios predichos

```
length.out = 2)
plot(aug.Pred, layout = c(4, 4, 1), # Fig. 8
key = list(lines = list(lty = c(1,2)),
text = list(c("Marginal", "Subject-specific")),
columns = 2))
```

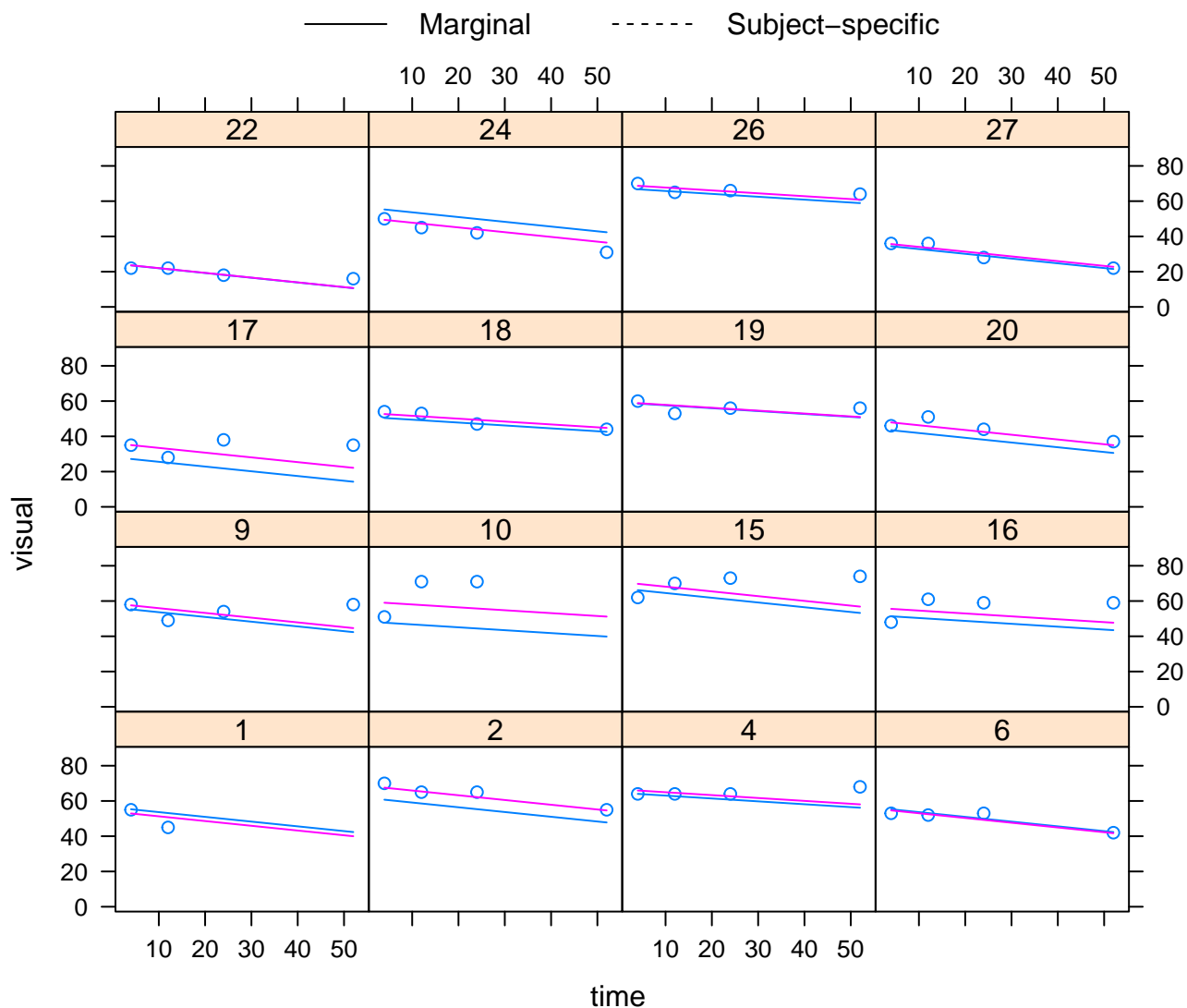


Figura 9: Valores observados y pronosticados de agudeza visual para pacientes seleccionados para el modelo 2

Establecemos `length.out = 2`, es decir, los valores pronosticados se obtienen en dos valores de tiempo, en el mínimo (4 semanas) y el máximo (52 semanas). Cada celda corresponde a un solo sujeto; así, se trazan las predicciones para los primeros 16 sujetos. Las medias de población predichas, que se muestran en la gráfica, disminuyen linealmente en el tiempo. Lo que indica que las medias de la población se desplazan para pacientes individuales mediante intercepciones aleatorias específicas del sujeto.

Sin embargo, para algunos pacientes, los perfiles individuales predichos obtenidos se desvían de los observados. Por ejemplo, para los sujetos 4 y 15, los patrones individuales predichos sugieren una disminución de la agudeza visual con el tiempo, mientras que los valores observados en realidad aumentan con el tiempo. Una posible forma de mejorar las predicciones individuales es permitir no solo las intercepciones aleatorias específicas del paciente, sino también las pendientes aleatorias específicas del paciente, como haremos en el siguiente modelo.

3.3. Modelos con intersecciones y pendientes aleatorias

3.3.1. Modelo con una matriz general D

Ajustamos el modelo siguiente `lmm3` usando la sintaxis `random = ~ 1 + time | subject` para especificar la estructura de efectos aleatorios, implicando que, para cada nivel de la variable de agrupación de sujetos, se debe considerar una intersección aleatoria y una pendiente aleatoria para el tiempo. También se presentan los intervalos de confianza al 95 %. Los resultados muestran un bajo valor estimado del coeficiente de correlación para los efectos aleatorios b_{0i} y b_{2i} , igual a 0.0662. El intervalo de confianza para el coeficiente de correlación sugiere que los dos efectos aleatorios pueden no estar correlacionados.

```
(lmm3 <- # modelo 3
update(lmm2,
random = ~1 + time | subject,
data = my.armd))
```

Linear mixed-effects model fit by REML

```
Data: my.armd
Log-restricted-likelihood: -1969.751
Fixed: lmm1.form
      (Intercept)          visual0          time          treat.fActive
      1.5686852          0.9471815         -0.1568511         -0.8307524
time:treat.fActive
      -0.1269461
```

Random effects:

```
Formula: ~1 + time | subject
Structure: General positive-definite, Log-Cholesky parametrization
      StdDev      Corr
(Intercept) 7.6102640 (Intr)
time         0.2409356 0.066
Residual     5.8289670
```

Variance function:

```
Structure: Power of variance covariate
Formula: ~time
Parameter estimates:
      power
0.04806768
Number of Observations: 534
Number of Groups: 144
```

```
getVarCov(lmm3, individual = "2") # D
```

Random effects variance covariance matrix

```
      (Intercept)      time
(Intercept)  57.9160 0.12130
time         0.1213 0.05805
Standard Deviations: 7.6103 0.24094
```

```
intervals(lmm3, which = "var-cov") # 95% CI para D, d, s
```

Approximate 95% confidence intervals

```
Random Effects:
Level: subject
      lower      est.      upper
sd((Intercept))  6.2437165 7.61026405 9.2759046
sd(time)         0.1927223 0.24093557 0.3012104
cor((Intercept),time) -0.3519550 0.06615641 0.4622586
```



```
Variance function:
      lower      est.      upper
power -0.06546967 0.04806768 0.161605
attr(,"label")
[1] "Variance function:"
```

```
Within-group standard error:
      lower      est.      upper
4.315751 5.828967 7.872757
```

3.3.2. Modelo con una Matriz Diagonal D

Creamos el modelo 4 lmm4 especificando `random = pdDiag (~ time)` de forma que implicamos una forma diagonal de la matriz de varianza-covarianza D de las intersecciones y pendientes aleatorias. Según los IC del 95 % para todos los parámetros del modelo 4, se sugiere que la estructura media podría simplificarse eliminando la interacción `time:treat.fActive`.

```
(lmm4 <- # modelo 4
update(lmm3,
random = list(subject = pdDiag(~time)), # Diagonal D
data = my.armd))
```

Linear mixed-effects model fit by REML

```
Data: my.armd
Log-restricted-likelihood: -1969.831
Fixed: lmm1.form
      (Intercept)      visual0      time      treat.fActive
      1.8303994      0.9422724     -0.1565141     -0.8273143
time:treat.fActive
      -0.1261758
```

Random effects:

```
Formula: ~time | subject
Structure: Diagonal
      (Intercept)      time Residual
StdDev:      7.740808 0.244554 5.787922
```

Variance function:

```
Structure: Power of variance covariate
Formula: ~time
Parameter estimates:
      power
0.04927641
Number of Observations: 534
Number of Groups: 144
```

```
intervals(lmm4) # 95% CI for b, qD, d, s
```

Approximate 95% confidence intervals

Fixed effects:

```
      lower      est.      upper
(Intercept) -3.8008311 1.8303994 7.46162989
visual0      0.8444896 0.9422724 1.04005520
time        -0.2333574 -0.1565141 -0.07967094
treat.fActive -3.9485434 -0.8273143 2.29391486
time:treat.fActive -0.2332879 -0.1261758 -0.01906384
```

```
attr(,"label")
[1] "Fixed effects:"

Random Effects:
Level: subject

              lower      est.      upper
sd((Intercept)) 6.5798974 7.740808 9.1065416
sd(time)         0.2008298 0.244554 0.2977977

Variance function:
              lower      est.      upper
power -0.07264037 0.04927641 0.1711932
attr(,"label")
[1] "Variance function:"

Within-group standard error:
      lower      est.      upper
4.197441 5.787922 7.981062
```

Aplicamos la función `anova()` a los modelo 3 (alternativo) y 4 (nulo). Ya que ambos modelos tienen la misma estructura media, el uso de la prueba LR basada en REML está justificado.

```
kable(anova(lmm4, lmm3)[,3:9]) # H0: d12=0 (modelo 4 modelo 3)
```

	df	AIC	BIC	logLik	Test	L.Ratio	p-value
lmm4	9	3957.663	3996.102	-1969.831		NA	NA
lmm3	10	3959.502	4002.212	-1969.751	1 vs 2	0.1609453	0.6882877

El resultado no es estadísticamente significativo al nivel de significancia del 5 %. Indica que, al suponer una estructura diagonal más simple de la matriz D, no empeoramos el ajuste del modelo. Lo mismo nos indican los valores AIC: el valor de 3959.502 para el modelo 3 es ligeramente mayor que el valor de 3957.663 para el modelo 4, lo que indica un ajuste ligeramente mejor del último modelo.

Como los valores de la diagonal de la matriz D y σ son positivos, nos indica que la función aumenta con el tiempo, lo que concuerda con la observación del análisis exploratorio.

Si observamos ahora los gráficos asociados al modelo, la figura 10 presenta los residuos condicionales de Pearson para el modelo 4. En comparación con la gráfica similar para el modelo 2 muestra menos residuos con un valor absoluto mayor que el percentil 97.5 de la distribución normal estándar.

```
library(lattice)

bwplot(resid(lmm4, type = "p") ~ time.f | treat.f,
data = my.armd, id = 0.05, adj = c(-0.3, 0.5 ), ylab= "residuals", grid = "h",
panel = function(...) {
  panel.bwplot(..., pch = "|")
  panel.xyplot(..., jitter.x = TRUE)})
```

La figura 11 muestra la gráfica Q-Q normal de los residuos condicionales de Pearson por tiempo para el modelo 4 comparable a la gráfica correspondiente para el modelo 2.

```
qqnorm(lmm4, ~resid(.) | time.f)
```

La figura 12 presenta las gráficas normal Q-Q de los efectos aleatorios predichos para el modelo 4. El primer gráfico muestra las intersecciones aleatorias, similar al modelo 2 y el segundo las pendientes aleatorias. Este último está ligeramente más cerca de una línea recta que el primero.

```
qqnorm(lmm4, ~resid(.) | time.f)
qqnorm(lmm4, ~ranef(.))
```

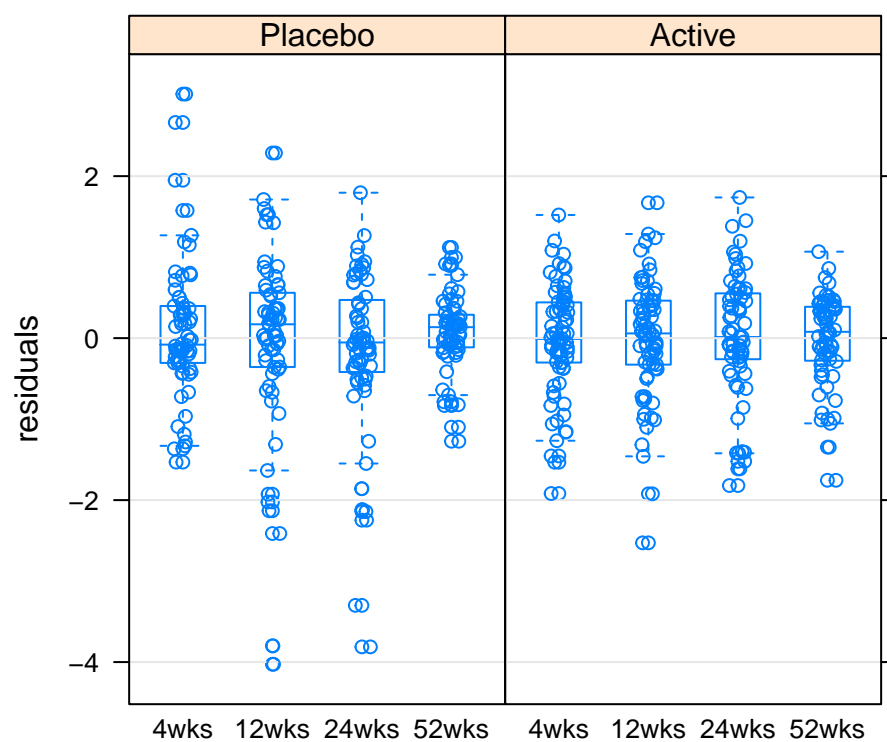


Figura 10: Stripplots de los residuos condicionales de Pearson para cada punto de tiempo y grupo de tratamiento para el modelo 4

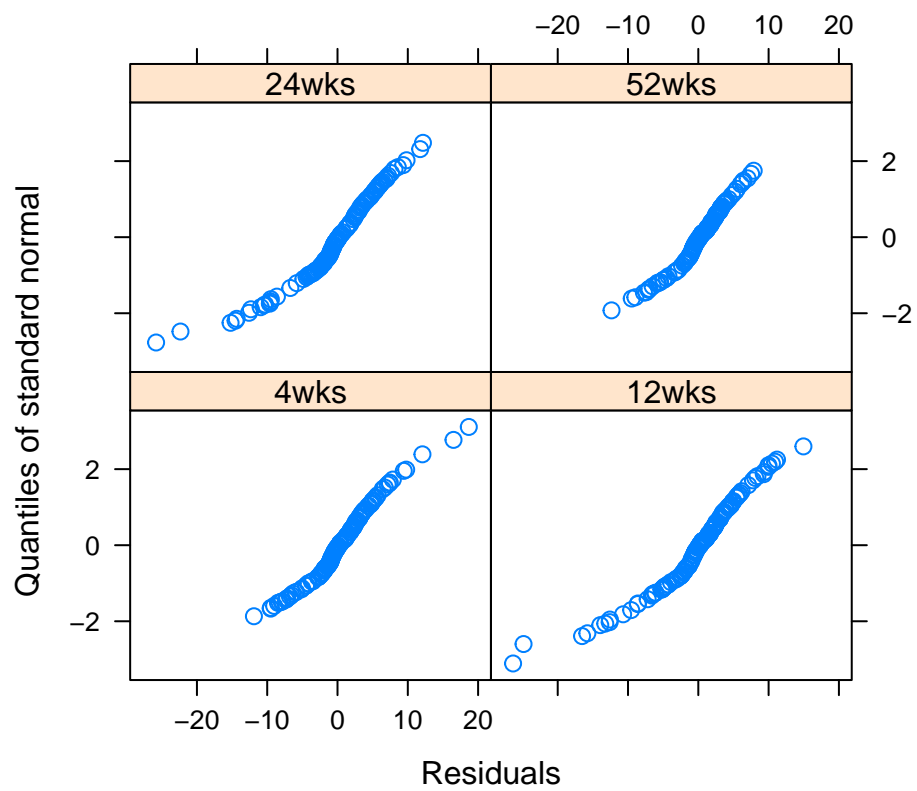


Figura 11: Gráficos Q-Q normales de residuos de Pearson

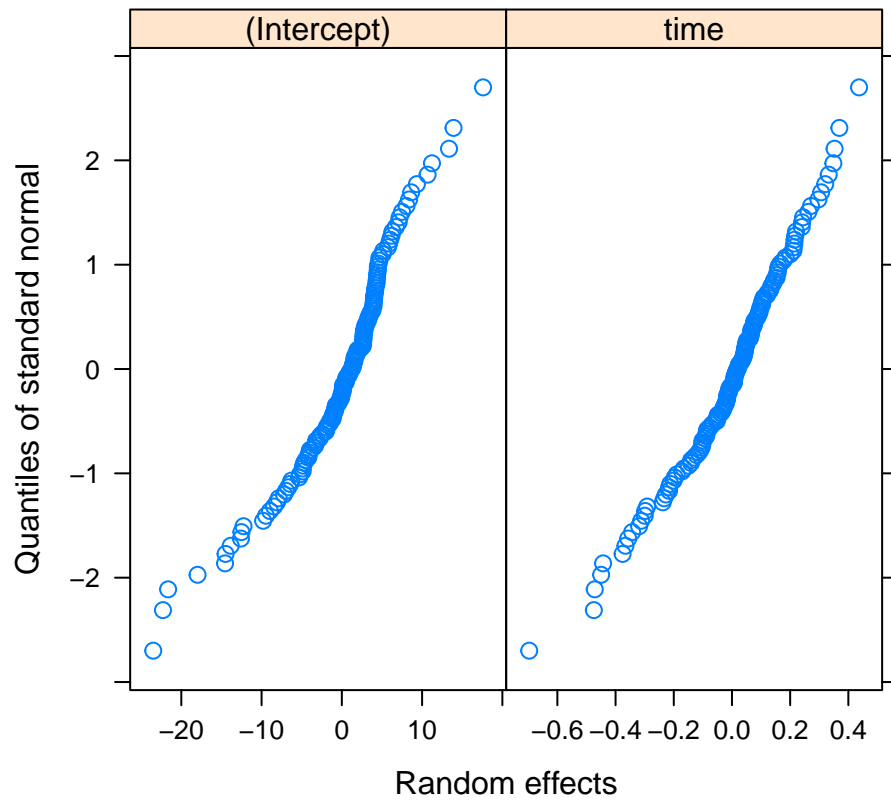
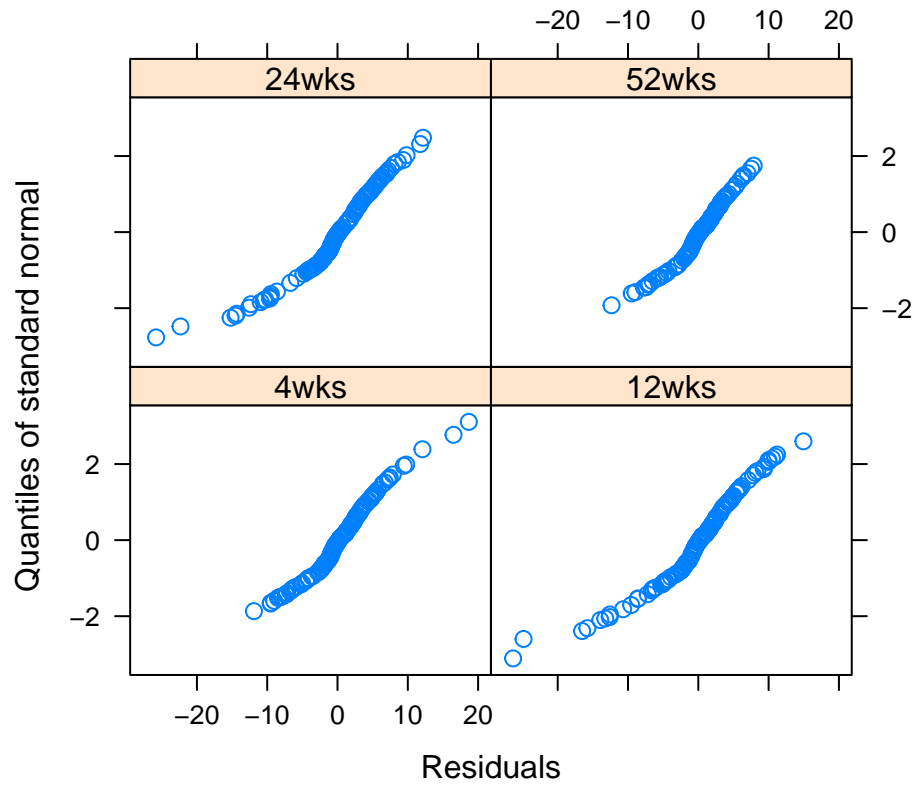


Figura 12: Gráficos Q-Q normales de efectos aleatorios predichos

Por último la figura 13 presenta los valores marginales y específicos del sujeto predichos para el modelo 4. En el modelo 2 la misma gráfica mostró una pendiente decreciente de los perfiles individuales, la misma para todos los sujetos y en algunos, los perfiles individuales predichos se desviaron fuertemente de los observados. En este modelo en cambio, los perfiles individuales predichos siguen más de cerca los valores observados y capturan, por ejemplo, las tendencias crecientes en el tiempo, lo que indica que **el modelo 4 ofrece un mejor ajuste a los datos que el modelo 2**.

```
aug.Pred <- # augPred aplicada al modelo 4
augPred(lmm4,
primary = ~time, # primera covariable
level = 0:1, # Marginal(0) and subj.-spec.(1)
length.out = 2)
plot(aug.Pred, layout = c(4, 4, 1),
key = list(lines = list(lty = c(1,2)),
text = list(c("Marginal", "Subject-specific")),
columns = 2))
```

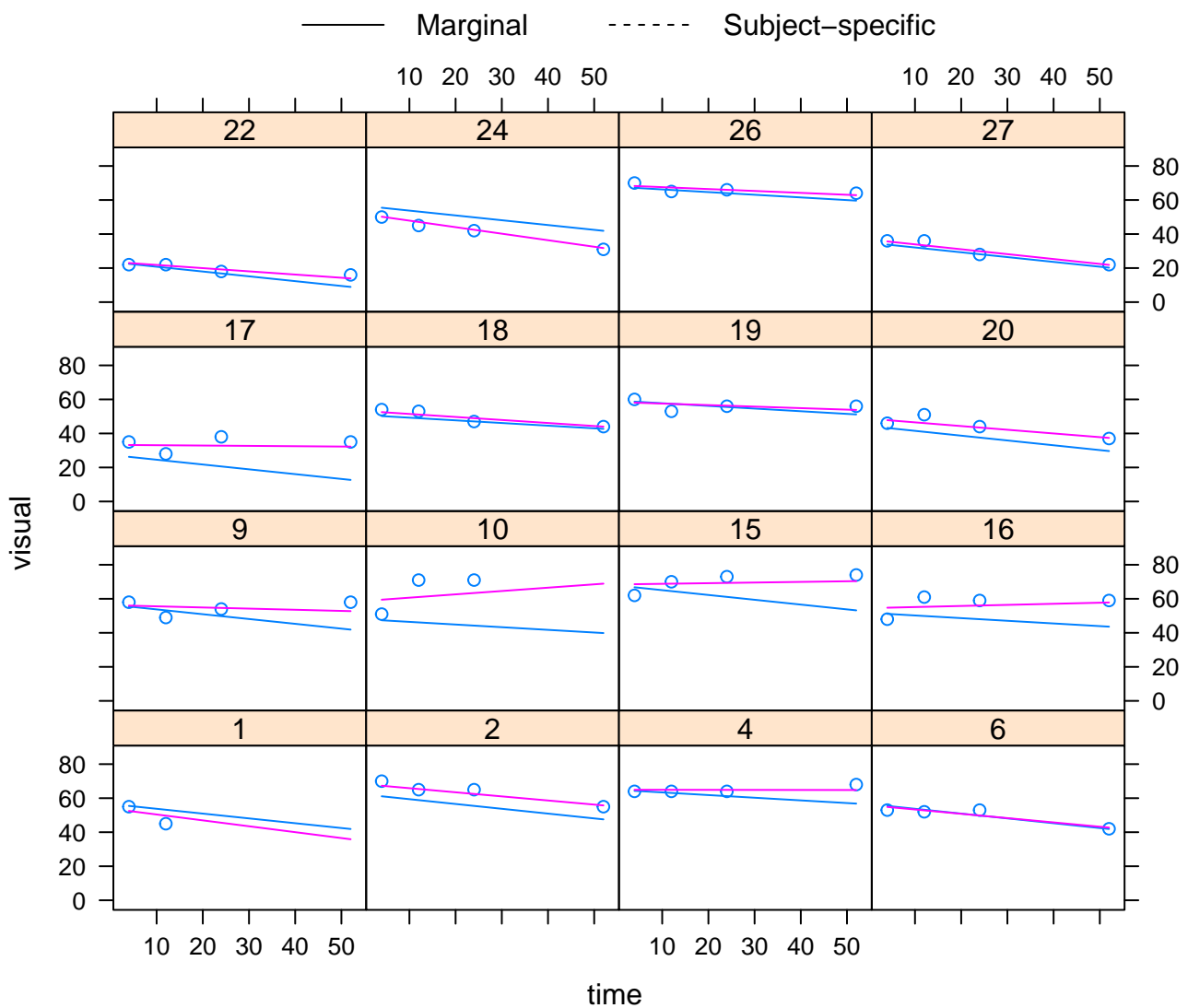


Figura 13: Valores observados y pronosticados de agudeza visual para pacientes seleccionados para el modelo 4

3.3.3. Modelo con una matriz diagonal D y un efecto de tratamiento constante

Como hemos visto en el apartado anterior, la estructura media del modelo 4 podría simplificarse eliminando la interacción `treat.f:time`. Para ajustar el nuevo modelo `lmm5` ajustamos la fórmula del modelo lineal (LM) y eliminamos la interacción, pero mantenemos el resto de variables.

```
lmm5.form <- formula(visual ~ visual0 + time + treat.f) # formula
lmm5 <- # modelo 5 a partir del 4
update(lmm4,
lmm5.form, data = my.armd)
lmm5
```

```
Linear mixed-effects model fit by REML
Data: my.armd
Log-restricted-likelihood: -1970.48
Fixed: visual ~ visual0 + time + treat.f
(Intercept)      visual0      time treat.fActive
  2.3252464      0.9427245    -0.2218854    -1.8170540
```

```
Random effects:
Formula: ~time | subject
Structure: Diagonal
(Intercept)      time Residual
StdDev:      7.729429 0.2492661 5.624642
```

```
Variance function:
Structure: Power of variance covariate
Formula: ~time
Parameter estimates:
power
0.06105921
Number of Observations: 534
Number of Groups: 144
```

```
summary(lmm5)$tTable #b, se(b), t-test
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	2.3252464	2.8507361	389	0.8156652	0.4151908
visual0	0.9427245	0.0493906	141	19.0871120	0.0000000
time	-0.2218854	0.0276348	389	-8.0291898	0.0000000
treat.fActive	-1.8170540	1.5160403	141	-1.1985526	0.2327125

```
# Estimaciones de efectos fijos, sus errores estándar aproximados e intervalos de
# confianza del 95% para los parámetros de varianza-covarianza del modelo 5.
intervals(lmm5, which = "var-cov") # 95% CI for qD, d, s
```

Approximate 95% confidence intervals

```
Random Effects:
Level: subject
lower      est.      upper
sd((Intercept)) 6.5703822 7.7294294 9.0929381
sd(time)        0.2047945 0.2492661 0.3033948
```

```
Variance function:
lower      est.      upper
power -0.07170841 0.06105921 0.1938268
```

```
attr(,"label")
[1] "Variance function:"
```

```
Within-group standard error:
  lower      est.      upper
3.970295 5.624642 7.968326
```

Como se puede observar en la salida de código anterior, el resultado de t-test para el factor `treat.f` es estadísticamente significativo al nivel de significancia del 5 %, lo que sugiere un efecto promedio negativo independiente del tiempo del tratamiento *activo*. Los IC son muy similares al modelo 4.

En el script siguiente se muestra la matriz de varianza-covarianza marginal estimada V_i que indica una tendencia creciente de variaciones de las mediciones de agudeza visual a lo largo del tiempo, mientras que la matriz de correlación correspondiente sugiere una correlación decreciente entre las mediciones obtenidas en puntos de tiempo más distantes. Por ahora el **modelo 5 parece ser el más apropiado** para trabajar con nuestro conjunto de datos.

Las estimaciones de las matrices D, Ri y Vi para el modelo 5.

```
VarCorr(lmm5) # D
```

```
subject = pdDiag(time)
      Variance StdDev
(Intercept) 59.74407850 7.7294294
time         0.06213359 0.2492661
Residual     31.63660292 5.6246425
```

```
getVarCov(lmm5, # Ri
type = "conditional", individual = "2")
```

```
subject 2
Conditional variance covariance matrix
      1      2      3      4
1 37.472 0.000 0.000 0.000
2 0.000 42.853 0.000 0.000
3 0.000 0.000 46.638 0.000
4 0.000 0.000 0.000 51.256
Standard Deviations: 6.1215 6.5462 6.8292 7.1593
```

```
(lmm5.cov <- # Vi
getVarCov(lmm5,
type = "marginal",
individual = "2"))
```

```
subject 2
Marginal variance covariance matrix
      1      2      3      4
1 98.211 62.726 65.709 72.668
2 62.726 111.540 77.639 98.515
3 65.709 77.639 142.170 137.290
4 72.668 98.515 137.290 279.010
Standard Deviations: 9.9101 10.561 11.924 16.704
```

```
cov2cor(lmm5.cov[[1]]) # Corr(Vi)
```

	1	2	3	4
1	1.0000000	0.5993059	0.5560827	0.438989
2	0.5993059	1.0000000	0.6165225	0.558434
3	0.5560827	0.6165225	1.0000000	0.689309
4	0.4389890	0.5584340	0.6893090	1.000000

Para resumir, hemos contruido una serie de modelos con varias estructuras aleatorias: modelo 1 con intersecciones aleatorias y varianzas residuales homoscedásticas; modelo 2 con intersecciones aleatorias y variaciones residuales descritas por una función de variación en el tiempo; modelo 3 con intersecciones aleatorias correlacionadas y pendientes aleatorias y las variaciones residuales del tiempo y modelo 4 con intercepciones aleatorias independientes y pendientes aleatorias y las variaciones residuales de la potencia del tiempo. El último modelo dio un ajuste satisfactorio a los datos y nos permitió simplificar la estructura media al adoptar un efecto de tratamiento constante, como se refleja en el modelo 5.

Para resumir presento los resultados de los modelos 3, 4 y 5 en la siguiente tabla:

	Parámetro	lmm3	lmm4	lmm5
Log-REML value		-1969.751	-1969.831	-1970.48
<i>Fixed effects</i>				
Intercept	β_0	1.568	1.83	2.325
Visual acuity at t=0	β_1	0.947	0.942	0.943
Time (in weeks)	β_2	-0.157	-0.157	-0.222
Trt(Actv vs. Plcb)	β_3	-0.831	-0.827	-1.817
Tm \times Treat(Actv)	β_4	-0.127	-0.126	
<i>reStruct(subject)</i>				
SD(bi0)	\sqrt{d}	7.61(6.244-9.276)	7.741(6.57-9.10)	7.73(6.57-9.09)
SD(time)		0.241(0.193-0.301)	0.245(0.2-0.3)	0.25(0.2-0.3)
cor((Intercept),time)		0.066(-0.352,0.462)		
<i>Variance function</i>				
Power (TIMEd)	δ	0.048	0.049	0.061
Scale	σ	5.82(4.32-7.87)	5.79(4.2-8.0)	5.62(3.97-7.97)

3.4. Una función alternativa de varianza residual: `varIdent()`

Los modelos presentados hasta ahora se especificaron con el uso de la función de varianza `varPower()`. Sin embargo, esta puede ser una función demasiado limitada, porque supone que las variaciones de las mediciones de agudeza visual cambian como una función de potencia del tiempo de medición. Sin embargo, es posible que, una función de varianza más general, no restringida, permita obtener un mejor ajuste que la función anterior. Para ello creamos el modelo 6 y lo comparamos con el modelo 3.

Ajustar el modelo 6 y la función de varianza utilizando una prueba de razón de probabilidad basada en REML.

```
(lmm6 <-  
update(lmm3, weights = varIdent(form = ~1 | time.f)))
```

Linear mixed-effects model fit by REML

```
Data: my.armd  
Log-restricted-likelihood: -1962.695  
Fixed: lmm1.form  
      (Intercept)      visual0      time      treat.fActive  
      2.3193771      0.9350423     -0.1487329     -0.9475474  
time:treat.fActive  
      -0.1337217
```

Random effects:

```
Formula: ~1 + time | subject  
Structure: General positive-definite, Log-Cholesky parametrization  
      StdDev      Corr  
(Intercept) 7.7936199 (Intr)  
time         0.2800217 -0.141  
Residual    4.9375763
```

Variance function:

```

Structure: Different standard deviations per stratum
Formula: ~1 | time.f
Parameter estimates:
      4wks      12wks      24wks      52wks
1.0000000000 1.5184761310 1.5080578418 0.0003772822
Number of Observations: 534
Number of Groups: 144

```

```
kable(anova(lmm3, lmm6)[,3:9]) # función varPower() frente a la función varIdent()
```

	df	AIC	BIC	logLik	Test	L.Ratio	p-value
lmm3	10	3959.502	4002.212	-1969.751		NA	NA
lmm6	12	3949.389	4000.641	-1962.695	1 vs 2	14.11279	0.0008619

El resultado de la prueba es estadísticamente significativa al nivel de significación del 5% y sugiere que el uso de la función de varianza `varIdent()` más general para definir la matriz R_i , proporciona un mejor ajuste que el uso de `varPower()` (AIC del modelo 6 es ligeramente menor). Sin embargo, si miramos a los parámetros estimados, el valor del parámetro de la semana 52 es extremadamente pequeño (0.00037728) y difiere sustancialmente de los valores estimados de las semanas 4, 12 y 24. Esto es sorprendente, porque todos los análisis anteriores indicaron que la varianza de la última medición de agudeza visual (en la semana 52) fue la mayor. Además, si vemos los intervalos de confianza para los parámetros de varianza y covarianza, nos da el siguiente mensaje de error, lo que indica problemas con la estimación de la matriz de varianza-covarianza para las estimaciones de los parámetros:

```
intervals(lmm6, which = "var-cov")
```

Finalmente, el problema con la convergencia del algoritmo de estimación para el modelo 6 también se refleja en el gráfico QQ normal de los residuos condicionales de Pearson, donde se observa que los residuos para la semana 52 son todos iguales a 0. Esto significa que la estimación REML, con la función `lme()` no es un valor óptimo en el modelo 6.

```

# El panel durante la semana 52 indica el problema con el ajuste del modelo.
qqnorm(lmm6, ~resid(.)|time.f)

```

Puede deberse a que el modelo 6 tiene siete parámetros y dado que el número de parámetros es cercano al número de ecuaciones, puede producirse colinealidad entre los parámetros, con consecuencias en forma de problemas de convergencia del algoritmo de estimación. Por otro lado, el modelo 3 tiene menos parámetros que incluye una función de potencia de tiempo, que no es lineal. Por lo tanto, en este caso, es menos probable que aparezca la colinealidad. de esta forma, el modelo 3 impone una restricción adicional en la forma de la estructura marginal de varianza-covarianza, que parece darnos un modelo más apropiado.

3.5. Prueba de hipótesis sobre efectos aleatorios

Calculamos los valores de AIC de los modelos 1 al 5. Se puede ver que, por ejemplo, el AIC para el modelo 2, es decir, 4003.693, es mayor que el valor de 3949.827 para el modelo 3, lo que indica un mejor ajuste del último modelo. Además, como vimos en la figura 13, los valores predichos obtenidos para el modelo 3 están más de cerca los observados, en comparación con el modelo 2. El valor más bajo de AIC se obtiene para el modelo 4, lo que sugiere que el modelo proporciona el mejor ajuste general a los datos.

Por otro lado, los modelos 4 y 5, son idénticos en cuanto a sus efectos fijos y sólo varían en sus efectos aleatorios. Para comparar los dos modelos con los mismos efectos fijos pero con distintos efectos aleatorios, usamos un test de verosimilitud.

El AIC sugiere que el modelo 4 es mas apropiado, pero el BIC sugiere que el modelo 5 es mejor. El p-valor del test de verosimilitud indica que el modelo más complejo (5) es el más parsimonioso y por tanto elegiríamos éste.

```

# valores de Akaike's Information Criterion para los modelos 1 al 5
AIC(lmm1, lmm2,
lmm3, lmm4)

```

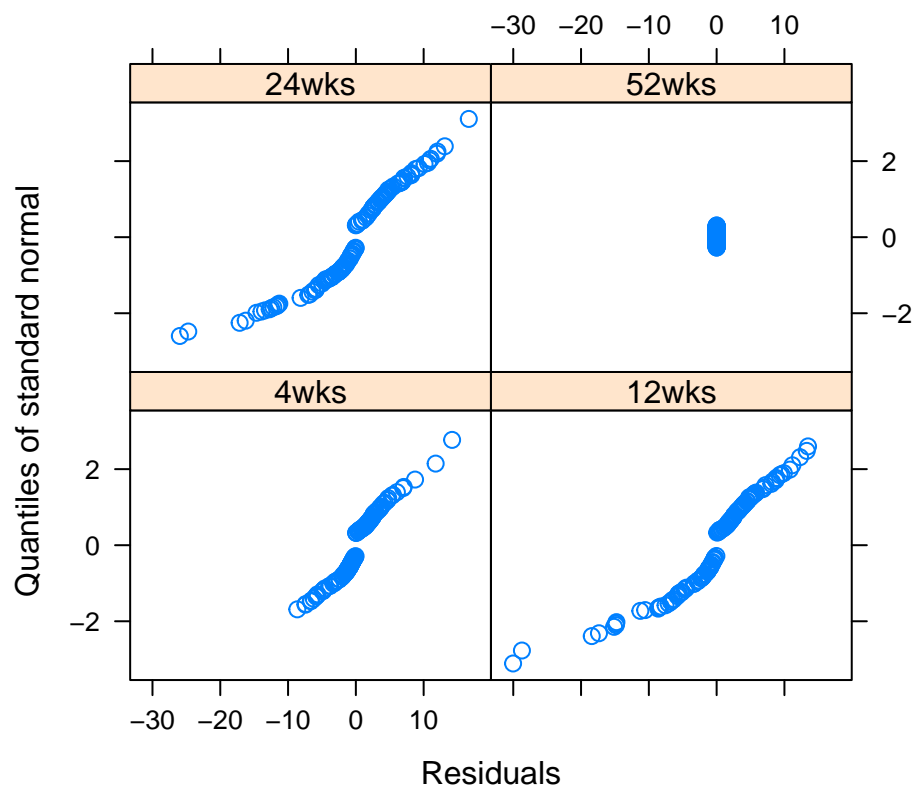


Figura 14: La gráfica Q-Q normal de los residuos condicionales de Pearson para modelo 6

	df	AIC
lmm1	7	4023.648
lmm2	8	4003.693
lmm3	10	3959.502
lmm4	9	3957.663

```
lmm4.ml <- update(lmm4, method = "ML")
lmm5.ml <- update(lmm5, method = "ML")
kable(anova(lmm4.ml, lmm5.ml)[,3:9]) # modelo 4 y 5
```

	df	AIC	BIC	logLik	Test	L.Ratio	p-value
lmm4.ml	9	3948.027	3986.551	-1965.014		NA	NA
lmm5.ml	8	3951.338	3985.581	-1967.669	1 vs 2	5.310996	0.0211912

3.5.1. Prueba de intercepciones aleatorias

Usando el modelo 1, que contiene intercepción aleatoria, vamos a comprobar si se necesitan intercepciones aleatorias específicas del sujeto a través de una prueba anova comparando el modelo 1 y un modelo nulo basado en el modelo 1, que asume errores residuales homoscedásticos y ningún efecto aleatorio. Básicamente estamos probando la hipótesis nula de que la varianza de la intersección aleatoria es cero, que está en el límite del espacio de parámetros.

Para obtener el valor p correcto, dividimos el valor χ^2 based p-value extraído del objeto `anova.res` entre 2. El p-value obtenido indica que el resultado de la prueba es estadísticamente significativo. Por ello, rechazamos la hipótesis nula de que la varianza de la distribución de las intersecciones aleatorias es igual a 0.

Empleamos también una alternativa usando el paquete `RLRsim`. El resultado muestra que el p-value de la prueba LR basada en REML, estimado a partir de 10,000 simulaciones, es estadísticamente significativo.

```
# 0.5 chi2_0 + 0.5chi_1 como la distribución nula

gls.1 <- # Null model
gls(lmm1.form, data = my.armd)
(anova.res <- anova(gls.1, lmm1)[,3:9]) # Null vs. modelo 1
```

	df	AIC	BIC	logLik	Test	L.Ratio	p-value
gls.1	6	4184.620	4210.246	-2086.310		NA	NA
lmm1	7	4023.648	4053.545	-2004.824	1 vs 2	162.9718	0

```
(anova.res[["p-value"]][2])/2 # 0.5c20 + 0.5c21
```

```
[1] 1.268699e-37
```

```
# empleamos la función exactRLRT() para simular la distribución nula
library(RLRsim)
exactRLRT(lmm1) # alternativa
```

simulated finite sample distribution of RLRT.

(p-value based on 10000 simulated values)

```
data:
RLRT = 162.97, p-value < 2.2e-16
```

3.5.2. Prueba de pendientes aleatorias

Consideramos ahora un modelo con intercepciones aleatorias y pendientes no correlacionadas específicas del sujeto y errores residuales homoscedásticos independientes y emplearemos la prueba LR basada en REML para probar si se necesitan pendientes aleatorias en un nuevo modelo 7, creados a partir del modelo 4. La prueba implica la comparación de dos modelos, modelo 1 (nulo) y modelo 7 (alternativa). Hay tres formas de realizar esta prueba:

```
# Creamos el modelo 7
lmm7 <-
update(lmm4, weights = NULL, # Constant resid. variance
data = my.armd)

# 0.5 chi2_0 + 0.5chi_1 como la distribución nula
(an.res <- # modelo 1 = null
stats::anova(lmm1, lmm7)[,3:9]) # modelo 7 = alternative
```

	df	AIC	BIC	logLik	Test	L.Ratio	p-value
lmm1	7	4023.648	4053.545	-2004.824		NA	NA
lmm7	8	3956.291	3990.459	-1970.146	1 vs 2	69.35665	0

```
(RLRT <- an.res[["L.Ratio"]][2]) # LR-test
```

```
[1] 69.35665
```

```
0.5*pchisq(RLRT, 1, lower.tail = FALSE) + 0.5*pchisq(RLRT, 2, lower.tail = FALSE)
```

```
[1] 4.759634e-16
```

```
# empleamos la función exactRLRT() para simular la distribución nula
mAux <- # modelo auxiliar con ...
update(lmm1, random = ~0 + time|subject, # ... pendientes aleatorias
data = my.armd)
exactRLRT(m = mAux,
m0 = lmm1, # null
mA = lmm7) # alternative
```

simulated finite sample distribution of RLRT.

(p-value based on 10000 simulated values)

data:

RLRT = 69.357, p-value < 2.2e-16

```
# empleamos la función simulate() para simular la distribución nula
lmm1.sim <- # null
simulate(lmm1, m2 = lmm7, nsim = 10000) # alternative
```

El valor p ajustado indica que el resultado de la prueba es estadísticamente significativa. Por lo tanto, la prueba nos permite rechazar la hipótesis nula de que la varianza de las pendientes aleatorias es igual a 0. Empleamos además la función `exactRLRT()`, que solo permite efectos aleatorios independientes. El valor p simulado es esencialmente igual a 0, lo que indica que la hipótesis nula puede ser rechazada.

Finalmente, la función `simulate()` se aplica para obtener una gráfica de valores p empíricos y nominales de la prueba LR.

```
plot(lmm1.sim, df = c(1, 2),
abline = c(0,1, lty=2))
```

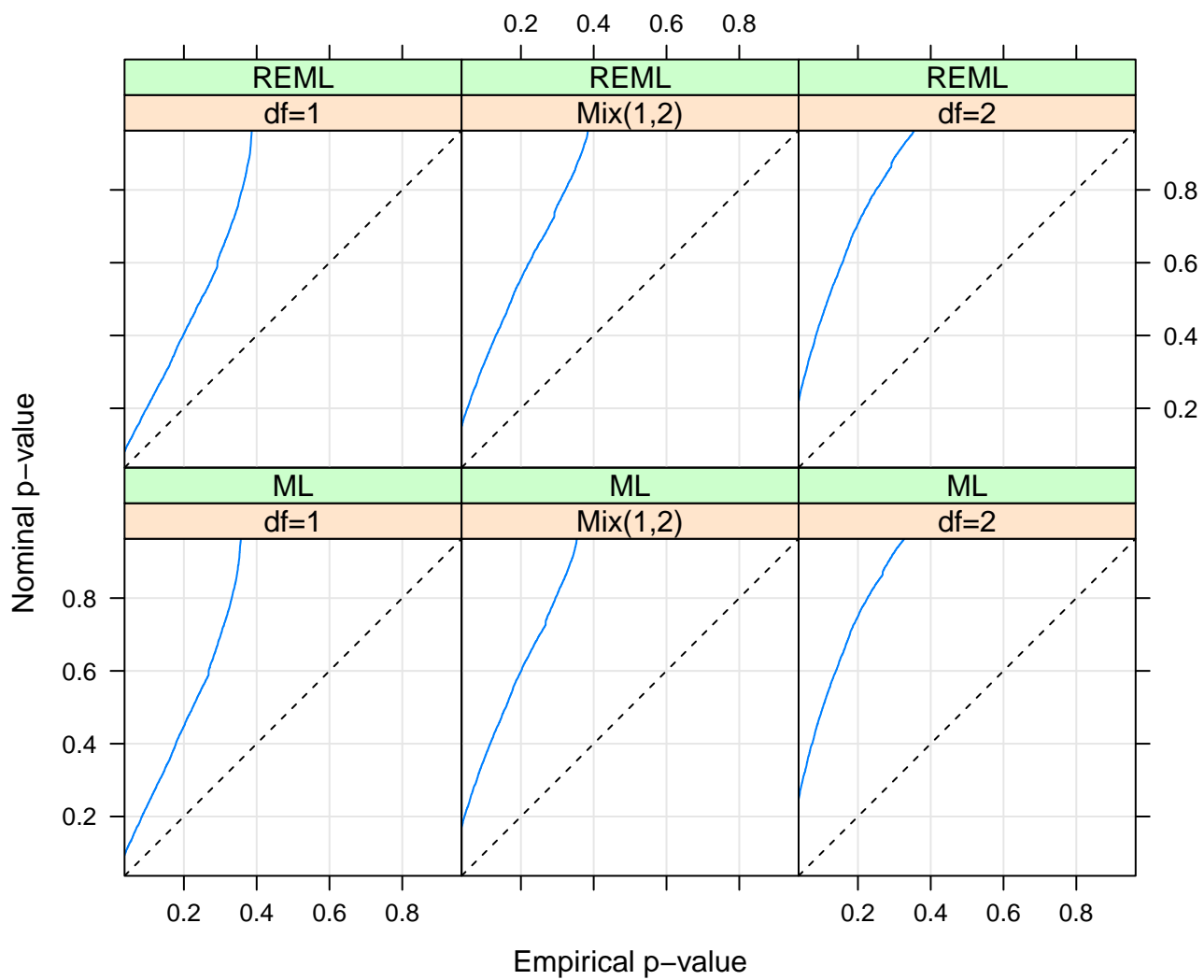


Figura 15: Valores p empíricos y nominales para probar la necesidad de pendientes aleatorias en el modelo 7

3.6. Análisis usando la función lmer()

Reajustamos los modelos 1 y 7, utilizando la función `lmer()` del paquete `lme4.0`.

Creamos el modelo con las mismas condiciones del modelo `lmm1.lmer` creado con la función `lme` del paquete `nlme()` y mostramos los resultados. Los valores de las estadísticas de la prueba t para los efectos fijos se proporcionan sin ningún valor de p en este caso. Además se muestra la matriz de varianza-covarianza del valor fijo.

```
# fijamos el modelo
library(lme4)
lmm1.lmer <-
lmer(visual ~ visual0 + time * treat.f + (1|subject),
data = my.armd)
print(lmm1.lmer, corr = FALSE) # % Corr(b)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: visual ~ visual0 + time * treat.f + (1 | subject)
Data: my.armd
REML criterion at convergence: 4009.648
Random effects:
Groups   Name             Std.Dev.
subject  (Intercept)  8.916
Residual                    8.219
Number of obs: 534, groups:  subject, 144
Fixed Effects:
              (Intercept)              visual0              time              treat.fActive
              5.1239              0.8782              -0.1502              -0.6319
time:treat.fActive
              -0.1207
```

```
# Mtriz de correlación
vcovb <- vcov(lmm1.lmer) # % Var(b)
corb <- cov2cor(vcovb) # % Corr( b)
nms <- abbreviate(names(fixef(lmm1.lmer)), 5)
rownames(corb) <- nms
corb
```

```
5 x 5 Matrix of class "dpoMatrix"
              (Intercept)              visual0              time              treat.fActive              time:treat.fActive
(Int)      1.0000000 -0.903489908 -0.197022898 -0.2329107 0.140929051
visl0      -0.9034899 1.000000000 -0.003269303 -0.0868464 0.002982792
time       -0.1970229 -0.003269303 1.000000000 0.3392377 -0.718205685
trt.A      -0.2329107 -0.086846400 0.339237743 1.0000000 -0.462087119
tm:.A      0.1409291 0.002982792 -0.718205685 -0.4620871 1.000000000
```

De la misma forma creamos el modelo `lmm7.lmer`:

```
library(lme4)
# Ajustamos el modelo y mostramos los resultados
lmm7.lmer <-
lmer(visual ~ visual0 + time + treat.f + treat.f:time +
(1|subject) + (0 + time|subject),
data = my.armd)
summ <- summary(lmm7.lmer)
coef(summ) # t-Table
```

	Estimate	Std. Error	t value
(Intercept)	1.7325232	2.8914454	0.5991893
visual0	0.9428910	0.0499312	18.8837886
time	-0.1541886	0.0387861	-3.9753576

	Estimate	Std. Error	t value
treat.fActive	-0.7493624	1.5956510	-0.4696280
time:treat.fActive	-0.1289571	0.0540730	-2.3848714

```
unlist(VarCorr(lmm7.lmer)) # D. Short printout
```

```
subject      subject.1
60.43472391  0.06026533
```

```
stats::sigma(lmm7.lmer) #s
```

```
[1] 6.560503
```

```
# Likelihood-ratio test para la interacción treat.f:time
```

```
lmm7lmer.aux <- # Model M16.7 with ...
```

```
update(lmm7.lmer, . ~ . - treat.f:time) #... interacción omitida
```

```
anova(lmm7lmer.aux, lmm7.lmer)
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
lmm7lmer.aux	7	3950.357	3980.320	-1968.178	3936.357	NA	NA	NA
lmm7.lmer	8	3946.683	3980.926	-1965.341	3930.683	5.674127	1	0.0172169

Por último, realizamos La prueba de razón de probabilidad basada en REML para pendientes no aleatorias en el modelo 7. Como vemos con valores y resultados similares.

```
# empleando 0.5 chi2_0 + 0.5chi_1 como la distribución nula
```

```
RML0 <- logLik(lmm1.lmer) # log-REML, (null)
```

```
RMLa <- logLik(lmm7.lmer) # log-REML, (alternative)
```

```
(RLRTstat <- -2 * as.numeric(RML0 - RMLa))
```

```
[1] 69.35665
```

```
# p-value
```

```
.5 * pchisq(RLRTstat, 1, lower.tail = FALSE) + .5 * pchisq(RLRTstat, 2, lower.tail = FALSE)
```

```
[1] 4.759634e-16
```

```
# empleamos la función exactRLRT() para simular la distribución nula.
```

```
require(RLRsim)
```

```
mAux <- lmer(visual ~ # modelo auxiliar con ...
```

```
visual0 + time + treat.f + treat.f:time +
```

```
(0 + time| subject), # ... pendientes aleatorias
```

```
data = my.armd)
```

```
exactRLRT(m = mAux, # modelo auxiliar
```

```
m0= lmm1.lmer, # (null)
```

```
mA= lmm7.lmer) # (alternative)
```

simulated finite sample distribution of RLRT.

(p-value based on 10000 simulated values)

```
data:
```

```
RLRT = 69.357, p-value < 2.2e-16
```


Bibliografia

Bates, Douglas, Martin Maechler, Ben Bolker, Steven Walker, Rune Haubo Bojesen Christensen, Henrik Singmann, Bin Dai, Gabor Grothendieck, Peter Green, and Maintainer Ben Bolker. 2015. “Package ‘lme4’.” *Convergence* 12 (1): 2.

Gałecki, Andrzej, and Tomasz Burzykowski. 2013. “Linear Mixed-Effects Model.” In *Linear Mixed-Effects Models Using R*, 245–73. Springer.

Kleinbaum, David G, and Mitchel Klein. 2010. *Survival Analysis*. Vol. 3. Springer.

Pinheiro, Jose, Douglas Bates, Saikat DebRoy, Deepayan Sarkar, R Core Team, and others. 2013. “Nlme: Linear and Nonlinear Mixed Effects Models.” *R Package Version 3* (1): 111.

Therneau, Terry M, and Thomas Lumley. 2014. “Package Survival.” *Survival Analysis Published on CRAN* 2: 3.