Modelización estadística avanzada con R: análisis de datos longitudinales

Juan R González

2021-10-05

Contents

Chapter 1

Preámbulo

Este bookdown sirve como notas para el curso Modelización estadística avanzada con R: análisis de datos longitudinales impartido en el Insituto Aragonés de Ciencias de la Salud

El contenido del curso tiene los siguientes temas:

- Módulo 1: Modelos lineales para datos longitudinales continuos (I)
  - Introducción
  - Formato de datos
  - Visualización de datos longitudinales
  - ANOVA para medidas repetidas
  - MANOVA
- Módulo 2: Modelos lineales para datos longitudinales continuos (II)
  - Modelos mixtos
- Módulo 3: Análisis de supervivencia con datos longitudinales
  - Análisis de supervivencia
  - Modelos con datos longitudinales
- Módulo 4: Modelización conjunta de datos longitudinales y de supervivencia
  - Introducción
  - Joint models

1.1 Instalación de librerías necesarias para el curso

Para poder reproducir todo el código de este libro se necesitan tener instaladas las siguientes librería

```
install.packages(c("tidyverse", "dplyr", "magrittr", "ggplot2", "reshape2",
                   "ez", "MANOVA.RM", "nlme", "ggeffects", "gridExtra",
```

```
                    "lme4", "chron", "compareGroups", "survminer", "JM"))
```

Los datos están accesibles en esta carpeta

`https://github.com/isglobal-brge/curso_longitudinal/tree/main/datos`

Chapter 2

Introducción a los diseños de datos longitudinales

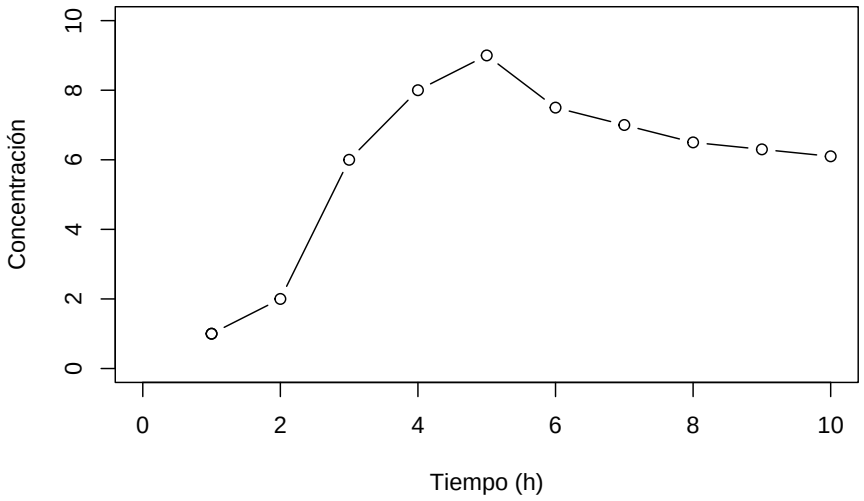
2.1 Definición

Un diseño longitudinal se obtiene cuando obtenemos distintas medidas de un sujeto a lo largo del tiempo. Por esto a veces a este tipo de datos también se les llama de medidas repetidas. En cada momento del tiempo se realiza una medida de una variable o de variables variables. Esta variable o variables diremos que serán cambiantes en el tiempo o tiempo-dependientes. En los diseños longitudinales típicamente se tiene una variable medida en distintos momentos del tiempo que será la variable respuesta (que normalmente queremos predecir o explicar) y opcionalmente otras variables que pueden medirse sólo en el momento basal o inicial o también pueden ser variables tiempo-dependientes.

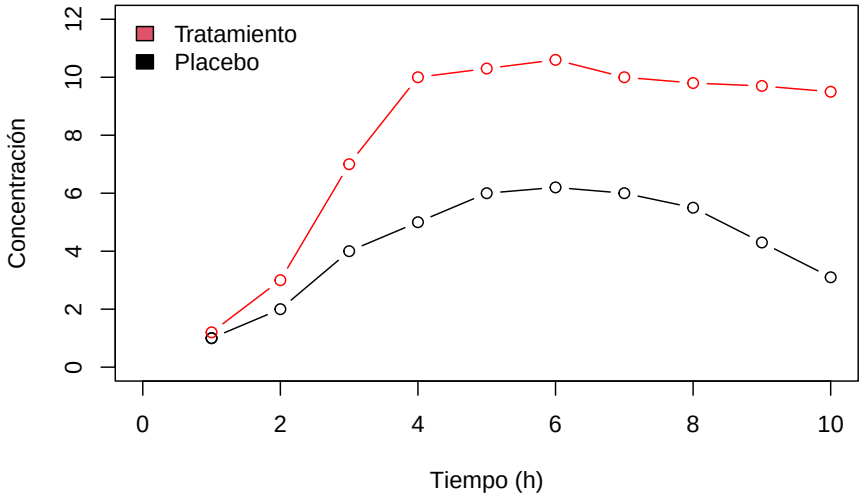
Hay variables que, aunque cambien en el tiempo, como puede ser la edad, como lo hace igual para todos, no se considerará tiempo-dependiente. Y otras que són constantes como el sexo.

Los **objetivos** pueden ser distintos:

- Estudiar la **evolución de una variable a lo largo del tiempo**. Esto es equivalente a evaluar el efecto que tiene el tiempo sobre esta variable.

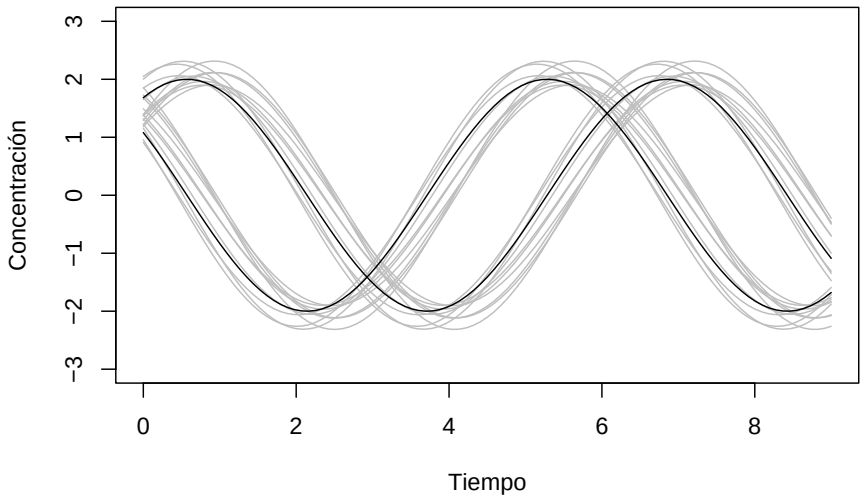


- Ver si la **evolución de una variable a lo largo del tiempo es la misma en diferentes grupos**. Por ejemplo ver si la concentración de un fármaco evoluaciona de la misma manera que otro fármaco ó placebo.



- **Identificar patrones en la evolución** o cambio de una variable a lo

largo del tiempo. En este contexto se pueden usar técnicas basadas en las distancias entre curvas (que no se explicarán en este curso).



2.2 Ejemplos

Ejemplo 1

En un estudio de intervención de dieta se reclutan 100 individuos por grupo. Estos grupos son (1) grupo control: se sigue una dieta mediterránea; (2) grupo de frutos secos: la dieta mediterranea es complementada con una cantidad de nueces; (3) grupo de aceite de oliva: a la dieta mediterránea se le añade una cantidad de aceite de oliva virgen. A lo largo de los siete años que dura el estudio, cada participante es visitado y se le toman distintas medidas desde cuestionarios de dieta, actividad física, medidas antropométricas (peso, talla), la tensión arterial, o medidas en sangre (perfil lipídico, etc.). No todos los individuos acuden en todas las visitas con lo cual aparecen datos faltantes. Además, por motivos de enfermedad o de muerte, algunos de alguno de ellos no se tiene la información de las visitas finales, dando lugar a distintos tiempos de seguimiento.

Ejemplo 2

A fin de comparar tres medicamentos (A, B y C) para reducir el colesterol, se reclutan 30 voluntarios, todos ellos diagnosticados de hipercolesterolemia y de entre 40 y 60 años de edad. Cada partipante

se le asigna uno de los tres medicamentos al azar de forma que 10 de ellos toman el medicamento A, otros el medicamento B y los 10 restantes el medicamento C.

Se miden los niveles de colesterol total en sangre (en mg/dL) justo en el inicio del estudio y cada 7 días en ayunas, a lo largo de 12 semanas.

Ejemplo 3

En un estudio sobre la polución ambiental en zonas urbanas, se eligen al azar 15 ciudades de más de 100.000 habitantes. De cada ciudad se eligen 4 puntos al azar con alta densidad de tráfico.

En cada punto se realizan 10 medidas desde las 8h de la mañana hasta las 17h de la tarde con intervalos de 60 minutos, en un día entre semana.

Ejemplo 4

Se quiere estudiar la población de una especie de alga en aguas marinas poco profundas. Para ello se muestrean veinte puntos al azar lo largo de la costa y a una distancia de aproximadamente un quilómetro de la playa. En cada punto se recoge una muestra a dos, cuatro, seis, ocho y diez metros de profundidas.

Finalmente, en cada muestra se contabilizan el número de especímenes que hay por centímetro cúbico de agua.

Ejemplo 5

En un estudio farmacocinético, se inyecta una cierta cantidad de un componente farmacológico en sangre. Se reclutan 15 individuos y para cada uno de ellos se mide cada hora la concentración en sangre.

Se puede estudiar las características de la curva de la concentración a lo largo del tiempo resumiendo todos los datos de concentración en un solo valor como es el AUC, o el tiempo de inflexión, etc. O bien, se puede analizar la concentración en cada punto desde el punto de vista de medidas repetidas.

Este ejemplo también se podría tratar como análisis funcional, viendo los datos de cada individuo como una función del tiempo.

Ejemplo 6

En un estudio sobre la adherencia a un medicamento, se hacen visitas mensuales a los participantes y en cada visita se pregunta si toma o no el medicamento con las dosis adecuadas, siendo la variable medida binaria (sí/no).

Ejemplo 7

En un estudio se quiere ver la eficacia de una dieta rica en aceite de oliva sobre los niveles de colesterol triglicéridos. Para ello se reculatan 3,000 participantes que son aleatorizados al grupo de tratamiento basado en una dieta enriquecida con aceite de oliva y a un grupo control a los que sólo se les indica que sigan una dieta saludable y pobre en grasas. Cada participante se visita anualmente durante sieta años en que se mide, entre otras variables los niveles de triglicéridos. A fin de asegurar o evaluar hasta qué punto los participantes siguen la dieta que les ha tocado, también se mide en cada visita un parámetro en sangre sensible a la cantidad de aceite de oliva ingerido.

No serían medidas repetidas:

1. Expresión génica de distintos genes. En este ejemplo cada gen sería una variable distinta; nos podría interesar comparar los niveles de expresión entre distintos genes. Éste sería un ejemplo de muchas variables respuesta.

2. Datos de seguimiento en el que se quiere estudiar la incidencia de diabetes. Para ello se hacen distintas visitas y se reporta en cada visita si el paciente es diabético o no. En este ejemplo, cada medida se toma o se mide si y solo si en la medida anterior el resultado es negativo. Sería un ejemplo de análisis de supervivencia con tiempo discreto.

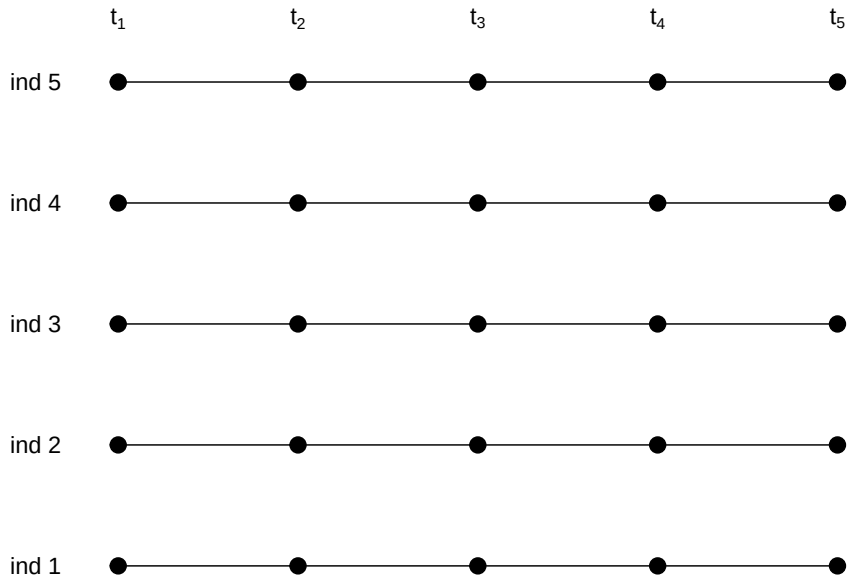
3. Si tenemos sólo dos medidas, aunque estrictamente son medidas repetidas, se pueden usar técnicas y modelos estándard. Por ejemplo, si la variable es continua se puede trabajar con la diferencia (después - antes) como la varaible respuesta y ajustar por el valor basal.

4. Si tenemos distintas medidas pero no sabemos cuando o en qué orden se han recogido. Por ejemplo, si tenemos 3 medidas de tensión arterial para cada individuo pero no sabemos el orden en que se han tomado las medidas. En este caso, se trataría o analizaría como datos en clúster. Alternativamente se puede calcular la media para cada individuo y trabajar con modelo estándar con un sólo dato por individuo.

2.3 Esquemas de recogidas de datos

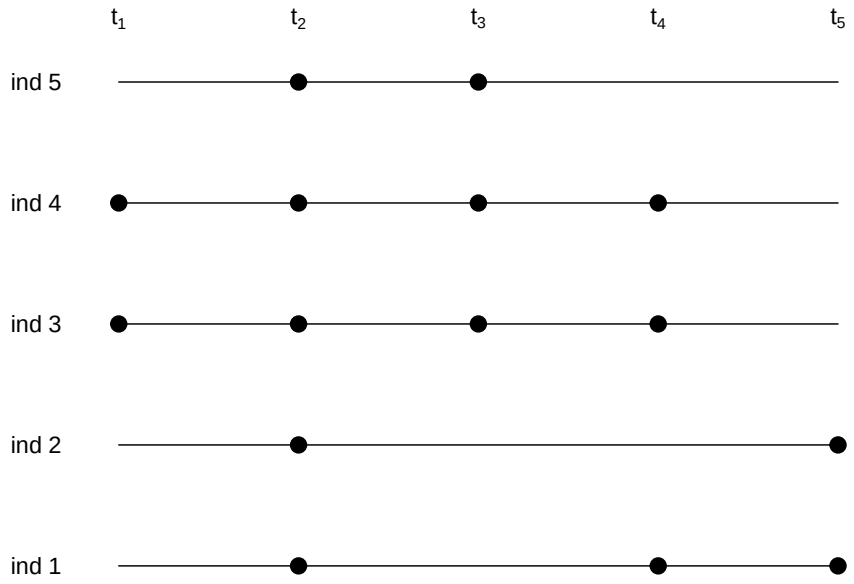
2.3.1 Todas las medidas con intervalos fijos

Medidas tomadas en los momentos t_1, t_2, ..., t_5. El intervalo transcurrido entre dos medidas puede ser constante o no. Pero son las mismas para todos los individuos. En caso que el intervalo no sea constante es importante anotar en qué momentos (segundos, minutos, horas, días, o metros...) se han hecho las medidas.



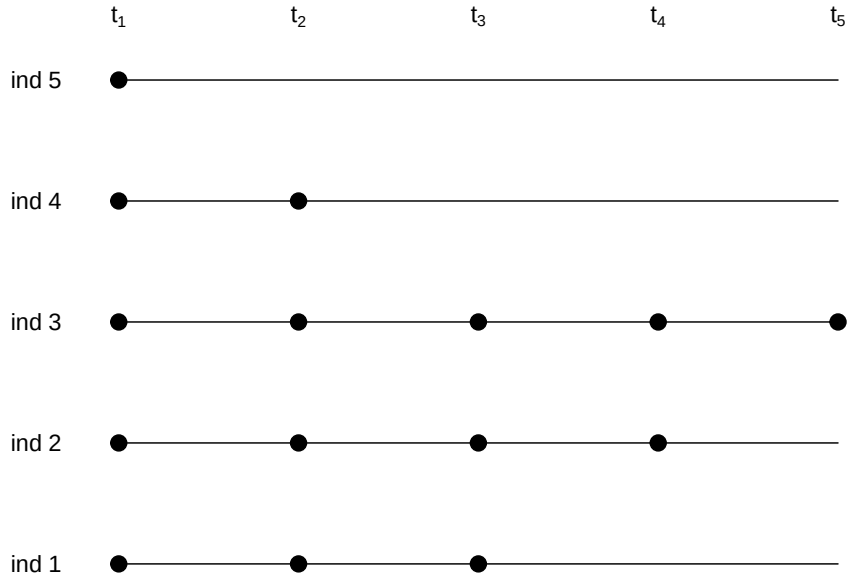
2.3.2 Missings al azar con intervalos fijos

No todos los individuos tienen datos observados en todos los tiempos. Es importante que estos datos faltantes ocurran al azar para poder considerar los modelos que vamos a ver en este curso como correctos.



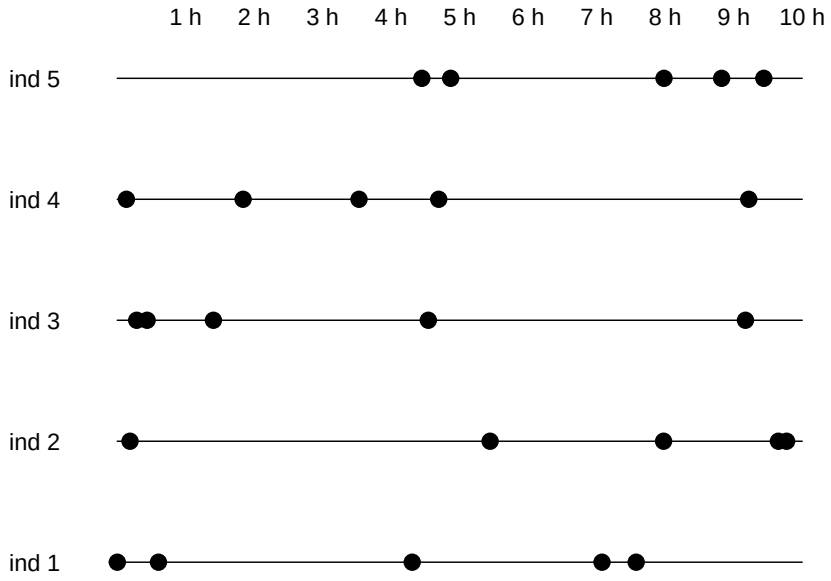
2.3.3 Distintos seguimientos

Algunos individuos son seguidos a lo largo del tiempo en más visitas que otros.



2.3.4 Todas las medidas a distintos tiempos

Los datos anteriores se consideran estudios de **datos panel**. Es decir, los datos se recogen de forma diara, semanal, anual, etc. (a tiempo fijo). Sin embargo, lo más habitual es disponer de observaciones en distintos tiempos para cada individuo y distintos número de observaciones para cada individuo.



Chapter 3

Estructura de los datos

3.1 Formato ancho y largo

Una vez que hemos recogido nuestros datos, debemos proceder con un análisis descriptivo para saber qué modelo debemos usar con el fin de contestar a las preguntas científicas planteadas en el estudio. La organización y visualización de los datos en estudios longitudinales no es tan sencilla como en cualquier otro tipo de diseño ya que los datos se pueden organizar de formas distintas. Básicamente, podemos:

- Datos a nivel de individuo, en el que cada individuo tiene un registro y múltiples variables contienen los datos de cada ocasión de medición (datos en formato ancho - *wide format* en inglés).
- Datos a nivel de tiempo, en el que cada persona tiene varios registros, uno para cada ocasión de medición (datos en formato largo - *long format* en inglés).

Las funciones que tenemos en R tanto para visualizar como modelizar datos longitudinales puede requerir el tener los datos en cualquiera de los dos formatos. El formato largo normalmente se necesita para agrupar variables como por ejemplo si queremos visualizar nuestra información mediante gráficos de barras apilados. En R existen muchas funciones para pasar de formato ancho a largo y viceversa. Recientemente se han creado las fuciones `dcast()` y `melt()` en la librería `reshape2` que facilitan enormemente estas tareas. La siguiente figura muestra un ejemplo de cómo utilizar estas funciones:

Veamos cómo realizarlo con R. Empecemos por cargar los datos que hemos visto en el ejemplo anterior

```
datos <- read.delim("datos/ejemplo.txt")
datos
```

Figure 3.1: Datos en formato ancho y largo y cómo pasar de un formato a otro usando funciones de la librería 'reshape2'

```
  id   sexo colesterol_1 colesterol_2
1  1 hombre          223          234
2  2 hombre          189          190
3  3  mujer          210          204
  colesterol_3
1          241
2          191
3          190
```

Vemos que están en formato ancho. Podemos pasarlos a formato largo utilizando la función `melt()` que tiene los siguientes argumentos:

- `data` es el objeto (data.frame o tibble) que vamos a convertir de ancho a largo,
- `id.vars` son las variables en la tabla que vamos a dejar sin cambiar de dimensión. En nuestro ejemplo sería la variable "sexo", aunque pueden ser más en tablas más complejas. Puede usarse un vector de nombres (tipo character) o de números enteros que correspondan al número de columna.
- `measure.vars` son las variables en las que se encuentran las mediciones. Puede ser un vector de nombres o de números enteros indicando los índices de las columnas. En nuestro caso son las columnas 3 a 5.
- `variable.name` es el nombre que va a adoptar la columna en la que queden nuestras variables, es decir, en nuestro caso sería el tiempo. Por defecto usa "variable".
- `value.name` es el nombre que va a adoptar la columna en la que queden los valores, que en nuestro caso sería el colestero. Por defecto usa "value".
- `variable.factor` es una valor lógico (TRUE o FALSE) para indicar si queremos que la columna de variable quede convertida a factor (opción por defecto), o quede simplemente como caracter.

Veamos cómo aplicamos esto a nuestro ejemplo

```
library(reshape2)
datos_largo <- melt(datos, measure.vars=3:5,
                    variable.name = "tiempo",
                    value.name = "colesterol")
datos_largo
```

```
  id   sexo       tiempo colesterol
1  1 hombre colesterol_1        223
2  2 hombre colesterol_1        189
3  3  mujer colesterol_1        210
4  1 hombre colesterol_2        234
5  2 hombre colesterol_2        190
6  3  mujer colesterol_2        204
7  1 hombre colesterol_3        241
8  2 hombre colesterol_3        191
9  3  mujer colesterol_3        190
```

Notemos que nuestra variable `tiempo` no es numérica indicando el momento donde se toma la medida de colesterol. Debería de ser una variable numérica 1, 2, 3. Podemos solucionar esto eliminando "colesterol_" de la variable simplemente ejecutando:

```
library(tidyverse)
datos_largo <- mutate(datos_largo,
                      tiempo = str_remove(tiempo, "colesterol_") %>%
                      as.numeric())
datos_largo
```

```
  id   sexo tiempo colesterol
1  1 hombre      1        223
2  2 hombre      1        189
3  3  mujer      1        210
4  1 hombre      2        234
5  2 hombre      2        190
6  3  mujer      2        204
7  1 hombre      3        241
8  2 hombre      3        191
9  3  mujer      3        190
```

Podemos ordenar nuestros datos por individuo y tiempo de la siguiente manera

```
datos_largo <- arrange(datos_largo, id, tiempo)
datos_largo
```

```
  id   sexo tiempo colesterol
1  1 hombre      1        223
2  1 hombre      2        234
3  1 hombre      3        241
4  2 hombre      1        189
5  2 hombre      2        190
6  2 hombre      3        191
7  3  mujer      1        210
8  3  mujer      2        204
9  3  mujer      3        190
```

Veamos ahora cómo pasar de formato largo a ancho. Para esta tarea usamos la función `dcast()`. Esta función tiene una notación un poco diferente, pues usa fórmulas para determinar qué variables poner en cada lugar. Tiene los siguientes argumentos:

- `data` es la tabla que vamos a convertir,
- `formula` es la forma en que vamos a distribuir las columnas. En general la fórmula es de forma x ~ y. Se puede usar una regla nemotécnica que consiste en: filas ~ columnas.
- `drop` deberían los valores faltantes ser eliminados o mantenidos?. Por

defecto es TRUE y no se ponen.
- `value.var` es el nombre (o número) de la columna en la que están los valores. Generalmente `dcast()` adivina bien este valor, pero es bueno usarlo para asegurarnos de lo que estamos haciendo y evitar que salga un mensaje de advertencia.

En nuestro caso ejecutaríamos:

```
datos_ancho <- dcast(datos_largo, id ~ tiempo,
                     value.var = "colesterol")
datos_ancho
```

```
  id   1   2   3
1  1 223 234 241
2  2 189 190 191
3  3 210 204 190
```

Si queremos mantener el resto de covariables debemos ejecutar:

```
datos_ancho <- dcast(datos_largo, id + sexo ~ tiempo,
                     value.var = "colesterol")
datos_ancho
```

```
  id   sexo   1   2   3
1  1 hombre 223 234 241
2  2 hombre 189 190 191
3  3  mujer 210 204 190
```

que es justo el conjunto de datos inicial del que partíamos excepto por el nombre de las variables (que ahora se llaman 1, 2, 3). Para poder poner el nombre original, basta con ejecutar:

```
datos_ancho <- dcast(datos_largo, id + sexo ~ paste0("colesterol_", tiempo),
                     value.var = "colesterol")
datos_ancho
```

```
  id   sexo colesterol_1 colesterol_2
1  1 hombre          223          234
2  2 hombre          189          190
3  3  mujer          210          204
  colesterol_3
1          241
2          191
3          190
```

Veamos ahora ejemplos más reales donde tenemos más de una variable repetida a lo largo del tiempo, datos faltantes u datos recogidos en distintos tiempos.

3.2 Formato ancho

Como hemos comentado anteriormente, lo normal es recoger los datos en formato ancho (u horizontal) data su simplicidad.

ind

sexo

edad

coltot_1

coltot_2

coltot_3

bmi_1

bmi_2

bmi_3

1

1

44

213

220

199

33.3

30.9

28.3

2

0

45

196

238

218

31.1

29.1

30.1

3

1
55
195
218
216
28.5
30.0
27.9
4
0
51
201
194
201
32.5
31.4
24.5
5
1
46
234
185
189
30.1
31.9
27.7
6
0
51
213
183

214

30.4

28.8

28.3

La **ventaja** de esta estrategia es que tenemos una fila para cada individuo, como estamos acostumbrados.

Sin embargo, existen varios **inconvenientes**:

- Si tenemos un missing en alguna medida hay que eliminar a todo el individuo
- Debemos suponer que todas las medidas se han realizado en los mismos momentos para todos los individuos, y esto puede no ser cierto.
- Las diferentes medidas de una misma variable predictora la debemos analizar como si fueran distintas variables

3.3 Formato largo

Estos mismos datos se dispondrían de la siguiente forma en formato largo (o vertical)

ind

sexo

edad

coltot

bmi

momento

1

1

44

213

33.3

1

1

1

44

220
30.9
2
1
1
44
199
28.3
3
2
0
45
196
31.1
1
2
0
45
238
29.1
2
2
0
45
218
30.1
3
3
1
55
195

28.5
1
3
1
55
218
30.0
2
3
1
55
216
27.9
3
4
0
51
201
32.5
1
4
0
51
194
31.4
2
4
0
51
201
24.5

3
5
1
46
234
30.1
1
5
1
46
185
31.9
2
5
1
46
189
27.7
3
6
0
51
213
30.4
1
6
0
51
183
28.8
2

6

0

51

214

28.3

3

3.4 Valores faltantes

Cuando hay valores faltantes en una medida y los datos se disponen de forma horizontal se descartan los demás valores ya que se elimina toda la fila.

ind

sexo

edad

coltot_1

coltot_2

coltot_3

bmi_1

bmi_2

bmi_3

1

1

44

213

220

199

33.3

30.9

28.3

2

0

45

196
NA
218
31.1
29.1
30.1
3
1
55
195
218
216
28.5
30.0
27.9
4
0
51
201
194
201
32.5
31.4
24.5
5
1
46
234
185
NA
30.1

31.9

27.7

6

0

51

213

183

214

30.4

28.8

28.3

En cambio, en la disposición vertical sólo se pierden los valores de los tiempos en cuestión y no todas las medidas del individuo.

ind

sexo

edad

coltot

bmi

momento

1

1

44

213

33.3

1

1

1

44

220

30.9

2

1
1
44
199
28.3
3
2
0
45
196
31.1
1
2
0
45
NA
29.1
2
2
0
45
218
30.1
3
3
1
55
195
28.5
1
3

1
55
218
30.0
2
3
1
55
216
27.9
3
4
0
51
201
32.5
1
4
0
51
194
31.4
2
4
0
51
201
24.5
3
5
1

46
234
30.1
1
5
1
46
185
31.9
2
5
1
46
NA
27.7
3
6
0
51
213
30.4
1
6
0
51
183
28.8
2
6
0
51

214

28.3

3

3.5 Tiempos de medidas diferentes

Al disponer los datos de forma vertical se puede especificar en qué momento se ha recogido cada medida. Para ello simplemente se indica en la variable tiempo. Por ejemplo si se trata de los días que han pasado desde el momento inicial del experimento.

ind

sexo

edad

coltot

bmi

momento

dias

1

1

44

213

33.3

1

1

1

1

44

220

30.9

2

4

1

1
44
199
28.3
3
5
2
0
45
196
31.1
1
3
2
0
45
238
29.1
2
7
2
0
45
218
30.1
3
10
3
1
55
195

28.5
1
2
3
1
55
218
30.0
2
7
3
1
55
216
27.9
3
8
4
0
51
201
32.5
1
2
4
0
51
194
31.4
2
8

4
0
51
201
24.5
3
9
5
1
46
234
30.1
1
1
5
1
46
185
31.9
2
9
5
1
46
189
27.7
3
10
6
0
51

213

30.4

1

3

6

0

51

183

28.8

2

6

6

0

51

214

28.3

3

10

O incluso podemos tener mas medidas para unos individuos que para otros. Como sería el caso que tuviéramos algún missing en alguna medida. Como en este ejemplo, para el individuo 2 y el 5 tenemos sólo 2 medidas, mientras que para el resto tenemos 3.

ind

sexo

edad

coltot

bmi

momento

dias

1

1

44

213
33.3
1
1
1
1
44
220
30.9
2
4
1
1
44
199
28.3
3
5
2
0
45
196
31.1
1
3
2
0
45
218
30.1
3

10
3
1
55
195
28.5
1
2
3
1
55
218
30.0
2
7
3
1
55
216
27.9
3
8
4
0
51
201
32.5
1
2
4
0

51
194
31.4
2
8
4
0
51
201
24.5
3
9
5
1
46
234
30.1
1
1
5
1
46
185
31.9
2
9
6
0
51
213
30.4

1

3

6

0

51

183

28.8

2

6

6

0

51

214

28.3

3

10

3.6 Transformación

3.6.1 Vertical a horizontal y viceversa

En esta sección aprovecharemos para ver otras instrucciones útiles en `R` para pasar de la disposición vertical de los datos a la horizontal y viceversa (aunque yo recomiendo usar `melt()` y `dcast()`. Para ello usaremos los datos del ejemplo anterior que están guardados en formato .csv de la siguiente manera:

```
tablahorizontal <- read.csv2("datos/tablahorizontal.csv")
tablahorizontal
```

```
  ind sexo edad coltot_1 coltot_2 coltot_3 bmi_1
1   1    1   44      213      220      199  33.3
2   2    0   45      196      238      218  31.1
3   3    1   55      195      218      216  28.5
4   4    0   51      201      194      201  32.5
5   5    1   46      234      185      189  30.1
6   6    0   51      213      183      214  30.4
  bmi_2 bmi_3
1  30.9  28.3
2  29.1  30.1
```

```
3  30.0  27.9
4  31.4  24.5
5  31.9  27.7
6  28.8  28.3
```

Como tenemos la base de datos en horizontal (una fila por individuo) y la queremos pasar a vertical (un registro por fila y varias filas por individuo) podemos usar:

```
tablong <- reshape(data=tablahorizontal,
                   direction="long",
                   varying=list(c("coltot_1","coltot_2","coltot_3"),
                                c("bmi_1","bmi_2","bmi_3")),
                   times=1:3,
                   timevar="momento",
                   idvar="ind",
                   v.names=c("coltot","bmi"))
tablong
```

```
    ind sexo edad momento coltot  bmi
1.1   1    1   44       1    213 33.3
2.1   2    0   45       1    196 31.1
3.1   3    1   55       1    195 28.5
4.1   4    0   51       1    201 32.5
5.1   5    1   46       1    234 30.1
6.1   6    0   51       1    213 30.4
1.2   1    1   44       2    220 30.9
2.2   2    0   45       2    238 29.1
3.2   3    1   55       2    218 30.0
4.2   4    0   51       2    194 31.4
5.2   5    1   46       2    185 31.9
6.2   6    0   51       2    183 28.8
1.3   1    1   44       3    199 28.3
2.3   2    0   45       3    218 30.1
3.3   3    1   55       3    216 27.9
4.3   4    0   51       3    201 24.5
5.3   5    1   46       3    189 27.7
6.3   6    0   51       3    214 28.3
```

Ordeno la tabla por id y dentro de cada id por tiempo

```
tablong <- arrange(tablong, ind, momento)
tablong
```

```
    ind sexo edad momento coltot  bmi
1.1   1    1   44       1    213 33.3
1.2   1    1   44       2    220 30.9
1.3   1    1   44       3    199 28.3
```

```
2.1   2    0   45       1    196 31.1
2.2   2    0   45       2    238 29.1
2.3   2    0   45       3    218 30.1
3.1   3    1   55       1    195 28.5
3.2   3    1   55       2    218 30.0
3.3   3    1   55       3    216 27.9
4.1   4    0   51       1    201 32.5
4.2   4    0   51       2    194 31.4
4.3   4    0   51       3    201 24.5
5.1   5    1   46       1    234 30.1
5.2   5    1   46       2    185 31.9
5.3   5    1   46       3    189 27.7
6.1   6    0   51       1    213 30.4
6.2   6    0   51       2    183 28.8
6.3   6    0   51       3    214 28.3
```

Y si queremos pasar del formato largo al horizontal

```
tablavertical <- read.csv2("datos/tablavertical.csv")
tablavertical
```

```
   ind sexo edad coltot  bmi momento dias
1    1    1   44    213 33.3       1    1
2    1    1   44    220 30.9       2    4
3    1    1   44    199 28.3       3    5
4    2    0   45    196 31.1       1    3
5    2    0   45    238 29.1       2    7
6    2    0   45    218 30.1       3   10
7    3    1   55    195 28.5       1    2
8    3    1   55    218 30.0       2    7
9    3    1   55    216 27.9       3    8
10   4    0   51    201 32.5       1    2
11   4    0   51    194 31.4       2    8
12   4    0   51    201 24.5       3    9
13   5    1   46    234 30.1       1    1
14   5    1   46    185 31.9       2    9
15   5    1   46    189 27.7       3   10
16   6    0   51    213 30.4       1    3
17   6    0   51    183 28.8       2    6
18   6    0   51    214 28.3       3   10
```

```
tabwide <- reshape(data=tablavertical,
                   direction="wide",
                   v.names=c("coltot","bmi"),
                   times=1:3,
                   timevar="momento",
                   idvar="ind")
```

```
tabwide
```

```
   ind sexo edad dias coltot.1 bmi.1 coltot.2
1    1    1   44    1      213  33.3      220
4    2    0   45    3      196  31.1      238
7    3    1   55    2      195  28.5      218
10   4    0   51    2      201  32.5      194
13   5    1   46    1      234  30.1      185
16   6    0   51    3      213  30.4      183
   bmi.2 coltot.3 bmi.3
1   30.9      199  28.3
4   29.1      218  30.1
7   30.0      216  27.9
10  31.4      201  24.5
13  31.9      189  27.7
16  28.8      214  28.3
```

¿Y si tenemos algun individuo con menos medidas? Por ejemplo, tenemos la tabla en formato vertical y para el individuo id=1 tenemos dos medidas en lugar de 3 (quitamos la tercera medida)

```
tablaverticalmiss <- tablavertical[-3,]
tablaverticalmiss
```

```
   ind sexo edad coltot  bmi momento dias
1    1    1   44    213 33.3       1    1
2    1    1   44    220 30.9       2    4
4    2    0   45    196 31.1       1    3
5    2    0   45    238 29.1       2    7
6    2    0   45    218 30.1       3   10
7    3    1   55    195 28.5       1    2
8    3    1   55    218 30.0       2    7
9    3    1   55    216 27.9       3    8
10   4    0   51    201 32.5       1    2
11   4    0   51    194 31.4       2    8
12   4    0   51    201 24.5       3    9
13   5    1   46    234 30.1       1    1
14   5    1   46    185 31.9       2    9
15   5    1   46    189 27.7       3   10
16   6    0   51    213 30.4       1    3
17   6    0   51    183 28.8       2    6
18   6    0   51    214 28.3       3   10
```

```
tabwidemiss <- reshape(data=tablaverticalmiss,
                       direction="wide",
                       v.names=c("coltot","bmi"),
                       times=1:3,
```

```
                      timevar="momento",
                      idvar="ind")
tabwidemiss
```

```
   ind sexo edad dias coltot.1 bmi.1 coltot.2
1    1    1   44    1      213  33.3      220
4    2    0   45    3      196  31.1      238
7    3    1   55    2      195  28.5      218
10   4    0   51    2      201  32.5      194
13   5    1   46    1      234  30.1      185
16   6    0   51    3      213  30.4      183
   bmi.2 coltot.3 bmi.3
1   30.9       NA    NA
4   29.1      218  30.1
7   30.0      216  27.9
10  31.4      201  24.5
13  31.9      189  27.7
16  28.8      214  28.3
```

3.6.2 Colapsar

Si tenemos los datos en vertical y queremos colapsar o resumir los distintos datos de cada individuo en un único valor, como por ejemplo la media.

```
library(dplyr)
library(magrittr)

group_by(tablavertical, ind) %>%
  summarise_at(vars(coltot, bmi), list(media = mean))
```

```
# A tibble: 6 x 3
    ind coltot_media bmi_media
  <int>        <dbl>     <dbl>
1     1         211.      30.8
2     2         217.      30.1
3     3         210.      28.8
4     4         199.      29.5
5     5         203.      29.9
6     6         203.      29.2
```

Chapter 4

Visualización de datos longitudinales

Para ilustrar cómo visualizar datos longitudinales usaremos los datos que se encuentran en formato ancho obtenidos de UCLA web site. Este ejemplo pertenece a un estudio realizado en adolescentes en el que se ha medido su tolerancia a tener un comportamiento que se "desvía de lo habitual" usando 9 ítems (medidos en una escala de 1 a 4 que va de un comportamiento muy malo hasta para nada malo) que se resumen con la media del valor obtenido en cada uno de ellos. Además del sexo del adolescente, también se ha recogido una variable ("exposure") que corresponde a la respuesta autoreportada por el adolescente a los 11 años de su exposición a tener un comportamiento que anómalo.

Los datos podemos cargarlos de la forma habitual, pero teniendo en cuenta que se encuentran en formato "csv". Notemos que no hace falta descargarlos en nuestro ordenador y cargarlos desde ahí, ya que la función `read_csv` acepta que los datos estén en un repositorio en la red (basta con indicar su URL).

```
library(tidyverse)
tolerance <- read_csv("https://stats.idre.ucla.edu/wp-content/uploads/2016/02/tolerance1.txt")
head(tolerance)
```

```
# A tibble: 6 x 8
     id tol11 tol12 tol13 tol14 tol15  male
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1     9  2.23  1.79  1.9   2.12  2.66     0
2    45  1.12  1.45  1.45  1.45  1.99     1
3   268  1.45  1.34  1.99  1.79  1.34     1
4   314  1.22  1.22  1.55  1.12  1.12     0
5   442  1.45  1.99  1.45  1.67  1.9      0
6   514  1.34  1.67  2.23  2.12  2.44     1
```

```
# ... with 1 more variable: exposure <dbl>
```

Con los datos en este formato, podemos aprovechar para obtener algunas estadísticas descriptivas que pueden resultar de interés. Por ejemplo, podemos ver cuántos individuos tenemos en nuestro estudio simplemente ejecutando:

```
nrow(tolerance)
```

```
[1] 16
```

4.1 Trayectorias

Para poder crear gráficos que nos informen sobre la evolución de la tolerancia entre individuos necesitamos que los datos estén en formato largo. Como hemos visto anteriormente, esto lo podemos realizar mediante

```
library(reshape2)
tolerance2 <- melt(tolerance, measure.vars=2:6,
                   variable.name = "age",
                   value.name = "tolerance")
# no nos olvidemos que nuestra variable edad
# debe de ser numérica
tolerance2 <- mutate(tolerance2,
                     age = str_remove(age,"tol")%>%
                        as.numeric())
head(tolerance2)
```

```
   id male exposure age tolerance
1   9    0     1.54  11      2.23
2  45    1     1.16  11      1.12
3 268    1     0.90  11      1.45
4 314    0     0.81  11      1.22
5 442    0     1.13  11      1.45
6 514    1     0.90  11      1.34
```

Puesto que nuestras observaciones para cada individuo se presenta a distintas edades, por eso hemos llamado `age` a nuestra variable temporal.

Con los datos en este formato no resulta tan sencillo saber cuántos individuos estamos analizando. Podemos usar `tidyverse` para obtener esta información

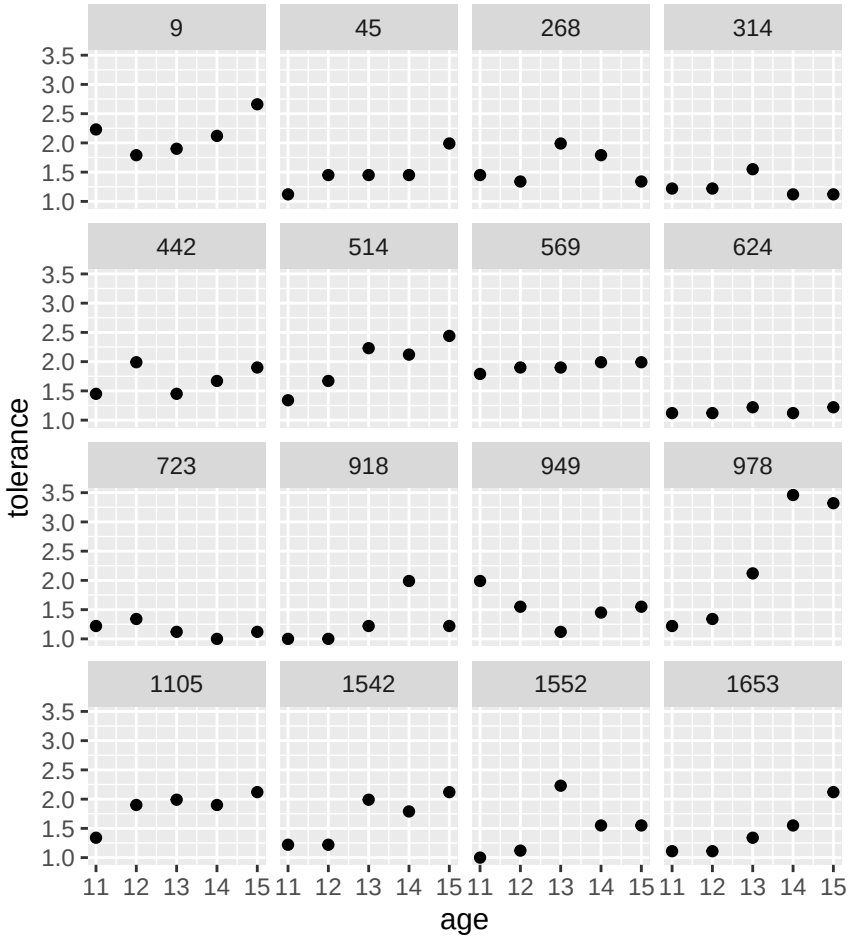
```
tolerance2 %>%
  distinct(id) %>%
  nrow()
```

```
[1] 16
```

NOTA: Para aquellos que no realizaron el último curso de R en el que se explicó tidyverse, se puede consultar este material.

Podemos empezar por visualizar nuestros datos creando lo que se conoce como *Empirical growth plots* o que nos nuestra la secuencia de nuestra variable de interés a lo largo del tiempo para cada individuo. En nuestro paso pondremos `age` en el eje X y `tolerance` en el Y. Para ello utilizaremos la función `geom_point()` y para crear el panel individual para cada sujeto usaremos `facet_wrap()` ambas son funciones de la librería `ggplot2`.

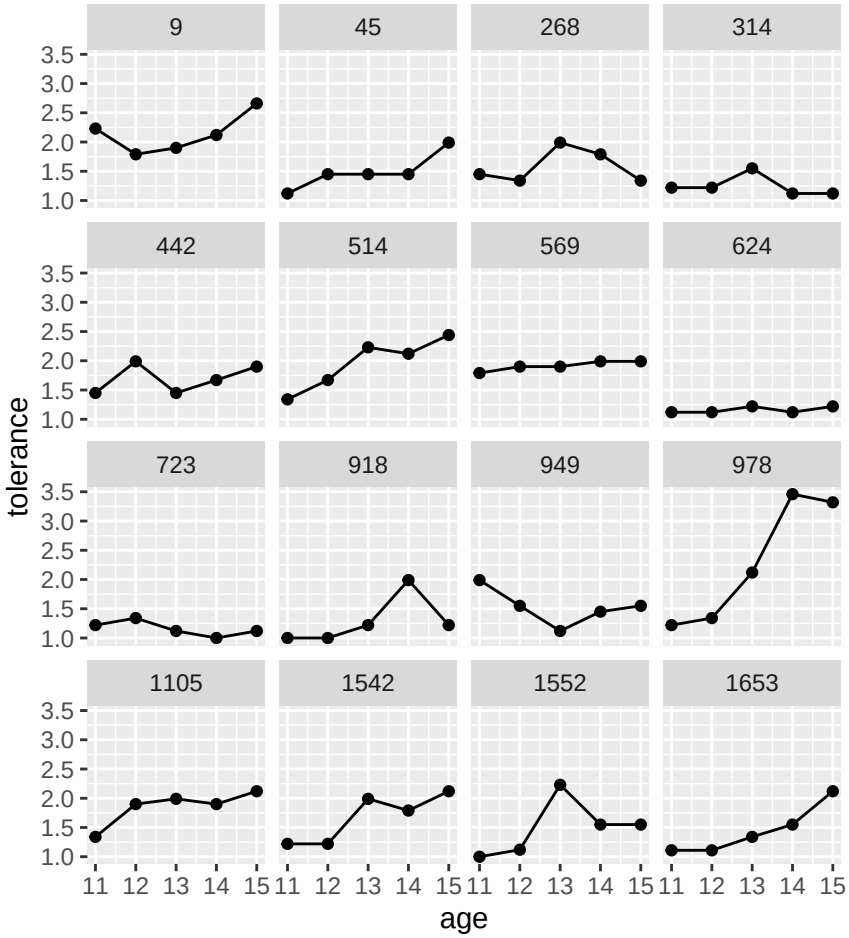
```
library(ggplot2)
ggplot(tolerance2, aes(x = age, y = tolerance)) +
  geom_point() +
  facet_wrap(~id)
```



Por defecto, `ggplot2` mantiene las escalas de ambos ejes iguales en todos los paneles. Si queremos que sea distinto entre cada individuo, podemos modificar el argumento `scales` en la función `facet_wrap()`.

Podemos añadir una línea para conectar los puntos usando `geom_line()`

```
ggplot(tolerance2, aes(x = age, y = tolerance)) +
  geom_point() +
  geom_line() +
  facet_wrap(~id)
```

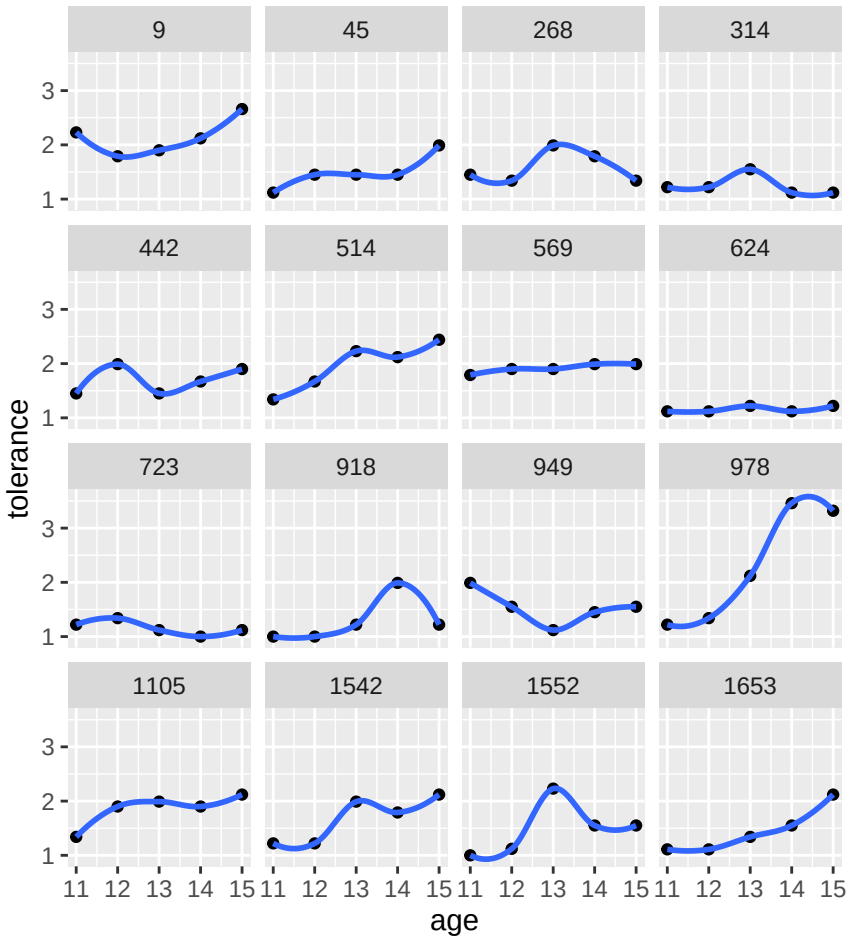


Sin embargo, a veces es recomendable utilizar otras aproximaciones que nos ayuden a visualizar mejor cuál es la trayectoria de cada individuo. Para ello, se puede utilizar otras aproximaciones como:

- suavizado no paramétrico
- funciones paramétricas

El suavizado paramétrico se puede llevar acabo usando un suavizado de tipo *loess*. Podemos usar esta opción mediante la función `stat_smooth()` y controlar el grado de suavizado con el argumento `span`.

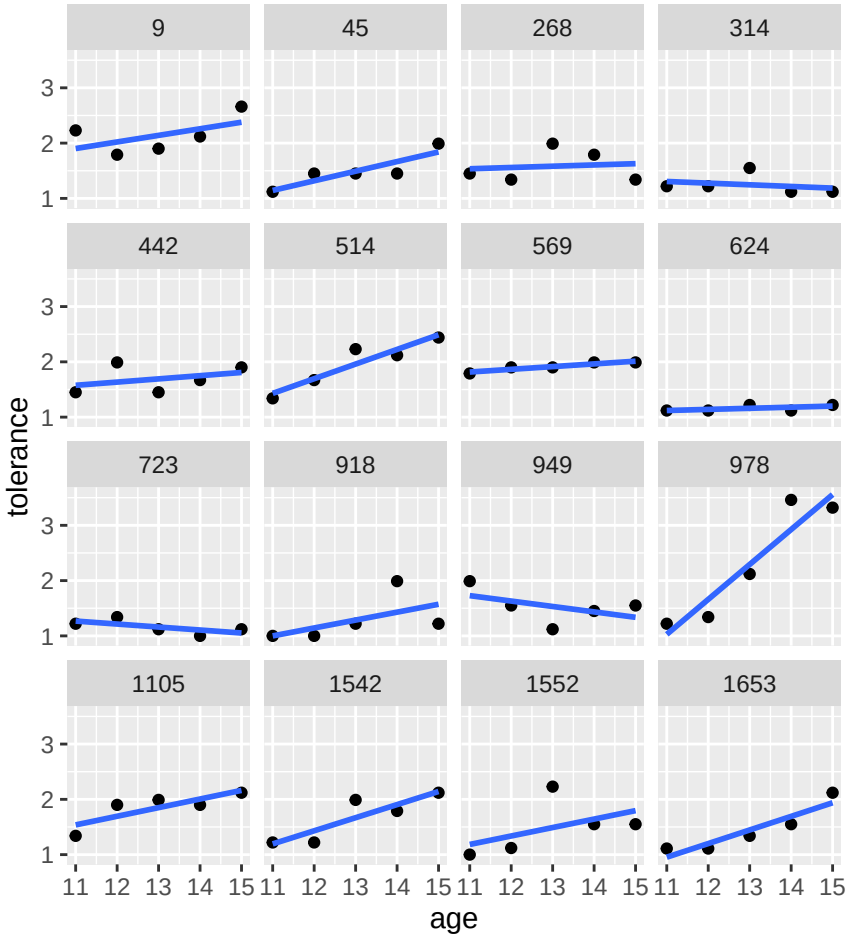
```
ggplot(tolerance2, aes(x = age, y = tolerance)) +
  geom_point() +
  stat_smooth(method = "loess", se = FALSE, span = .9) +
  facet_wrap(~id)
```



NOTA: El argumento `se=FALSE` sirve para que no pintemos la banda de confianza para la estimación no paramétrica.

Podemos visualizar las trayectorias usando un modelo paramétrico. El más sencillo sería un modelo lineal que podemos visualizar mediante el argumento `method = 'lm'`

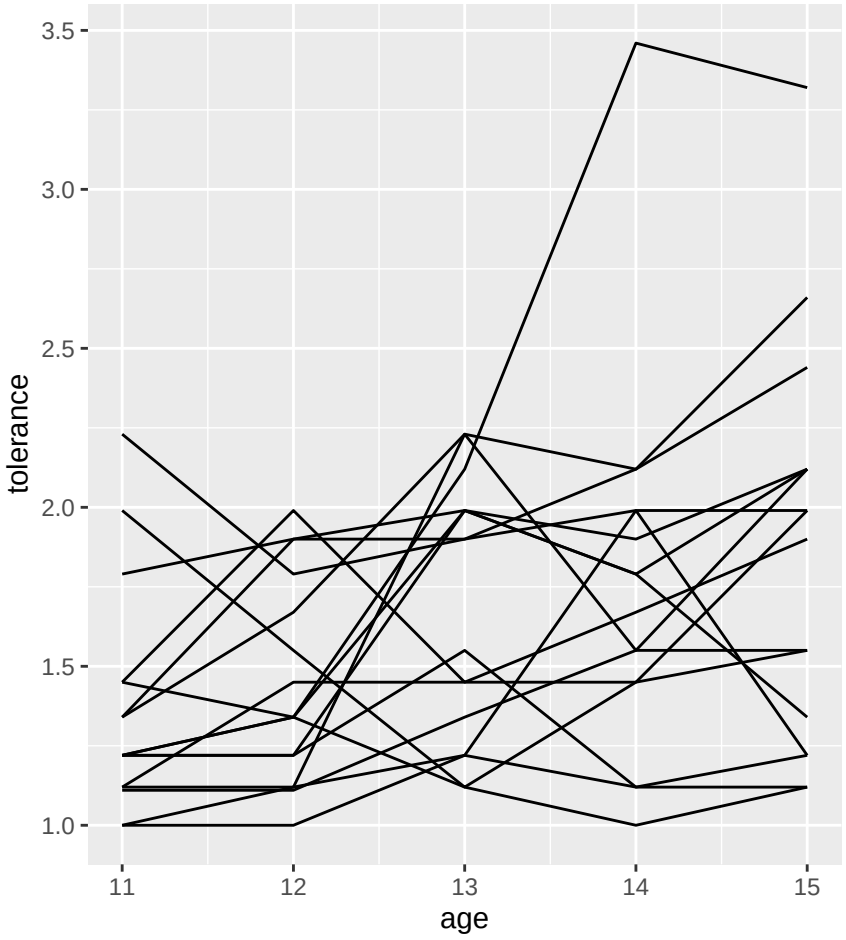
```
ggplot(tolerance2, aes(x = age, y = tolerance)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE) +
  facet_wrap(~id)
```



4.2 Spaghetti plots

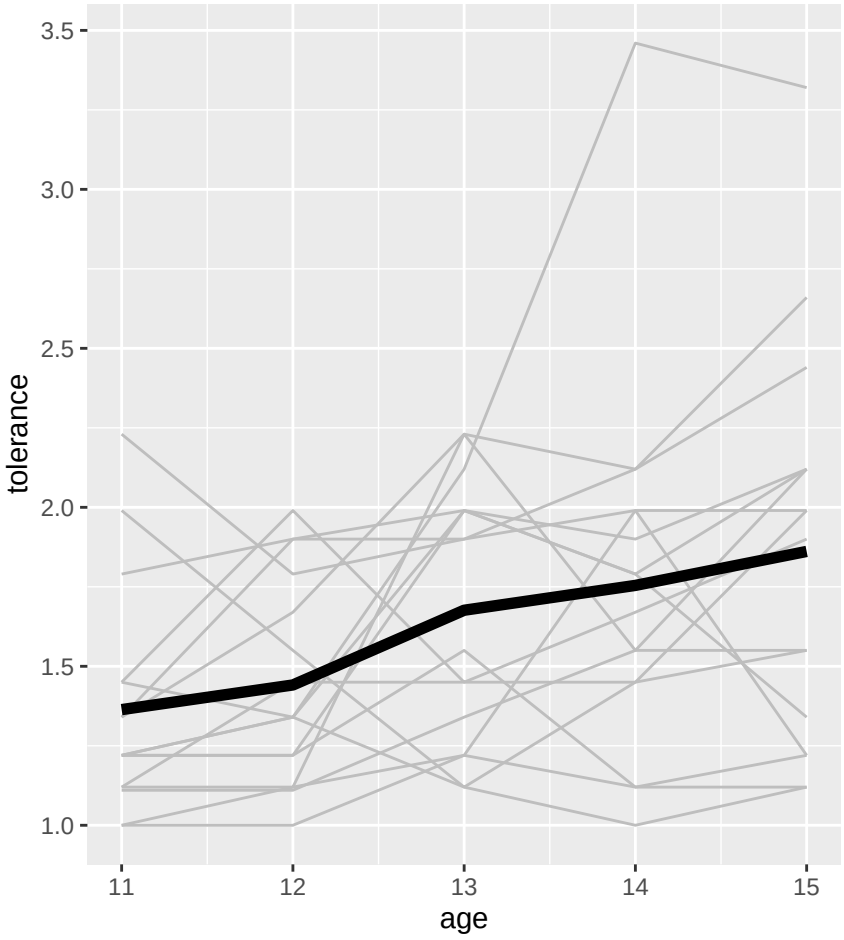
También podemos visualizar todas las trayectorias en único gráfico mediante lo que se conoce como un **spaghetti plot**. Esta representación es muy útil cuando queremos visualizar muchos individuos. Para ello deberíamos usar el argumento `group`

```
ggplot(tolerance2, aes(x = age, y = tolerance, group = id)) +
  geom_line()
```



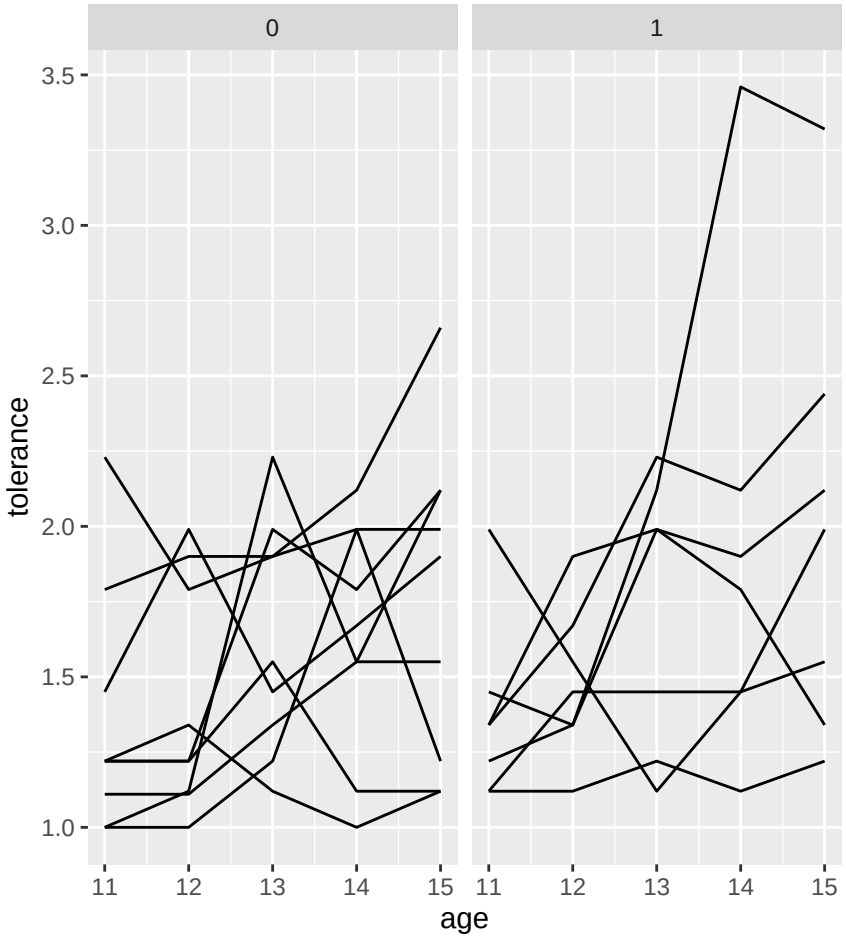
También podemos añadir el perfil promedio

```
ggplot(tolerance2, aes(x = age, y = tolerance, group = id)) +
  geom_line(col="grey") +
  stat_summary(aes(group = 1), geom = "line", fun = mean, size=2)
```



Podríamos obtener el mismo gráfico separado para hombres y mujeres usando de nuevo `facet_wrap()`

```
ggplot(tolerance2, aes(x = age, y = tolerance, group = id)) +
  geom_line() +
  facet_wrap(~male)
```

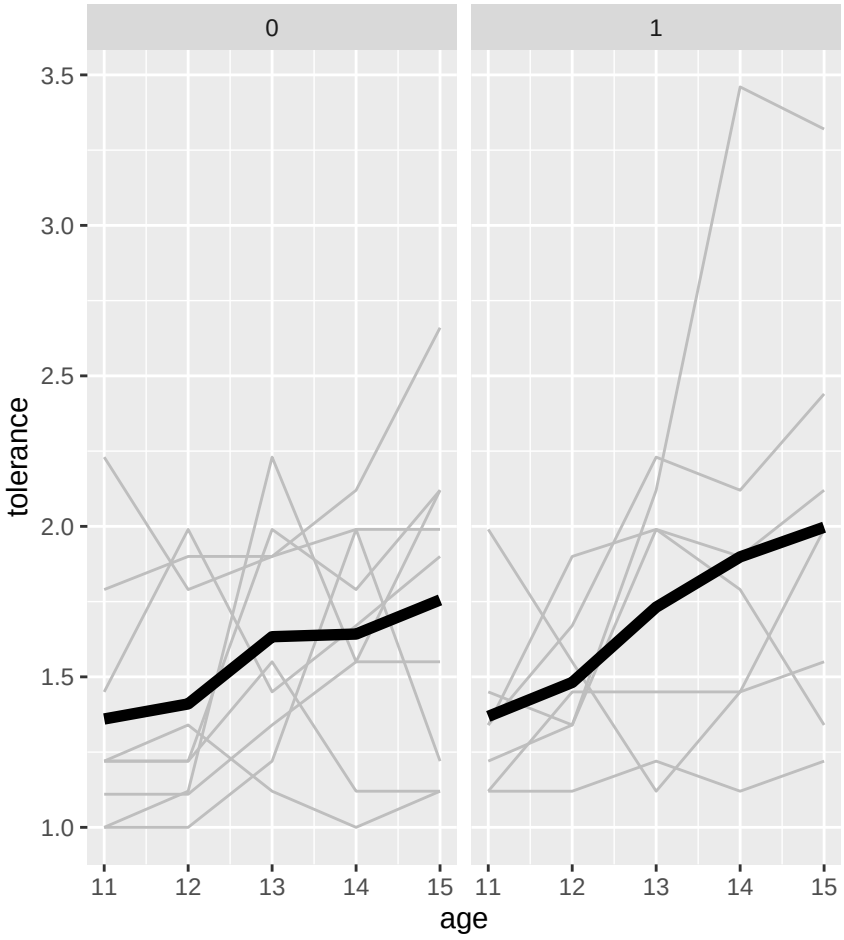


NOTA: En `ggplot2` podemos guardar un gráfico con un nombre y luego "reciclar" el gráfico añadiendo más código de la siguiente manera. Por ejemplo, puedo guardar el gráfico anterior en el objeto `p`

```
p <- ggplot(tolerance2, aes(x = age, y = tolerance, group = id)) +
  geom_line(col="gray") +
  facet_wrap(~male)
```

y luego decirle que me añada el perfir promedio:

```
p + stat_summary(aes(group = 1),
    geom = "line", fun = mean, size=2)
```


0
1
3.5
3.0
2.5
2.0
1.5
1.0
tolerance
11
12
13
14
15
age

Chapter 5

Modelos con respuesta normal

En este capítulo se describirán los métodos y modelos estadísticos para analizar medidas respetidas cuando la variable respuesta sigue una distribución normal o gaussiana. Se pueden probar algunas transformaciones, como el logaritmo, para normalizar la distribución de la variable.

5.1 Técnica de la suma de cuadrados

Este método o técnica se basa en la suma de cuadrados. Es la más simple desde el punto de vista estadístico y computacional. Por contra, sólo permite analizar diseños balanceados, sin variables independientes cuantitativas (covariables), sólo cualitativas o factores y con un número limitado de factores que tienen que estar cruzados (no anidados). A continuación se presentan los dos diseños más simples de medidas repetidas que se pueden analizar con esta técnica.

5.1.1 Diseño 1W+1B

Cuando el diseño es balanceado (mismo número de individuos por grupo), las medidas son las mismas para todos los individuos y no hay covariables, se puede usar la técnica de suma de cuadrados o tabla ANOVA.

La notación que se usa para la ecuación del modelo en el contexto de suma de cuadrados es:

$$y_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \pi_{k(i)} + e_{ijk}$$

Donde

- μ es la constante del modelo,
- α_i, son los efectos del grupo o tratamiento
- β_j, son los efectos del tiempo
- $\alpha\beta_{ij}$ es la interacción del tiempo con el grupo
- $\pi_{k(i)}$ es el efecto aleatorio del individuo que está anidado al grupo
- e_{ijk} son los errores

$\sum_{i=1}^{a} \alpha_i = 0$, $\sum_{j=1}^{b} \beta_j = 0$, $\sum_{i=1}^{a} \alpha\beta_{ij} = 0, \forall j$, $\sum_{j=1}^{b} \alpha\beta_{ij} = 0, \forall i$,

$\pi_{k(i)} \sim N(0, \sigma_{ind})$ $e_{ijk} \sim N(0, \sigma)$ indep

En este contexto se dice que el tiempo y la interacción tratamiento:tiempo son términos o componentes "intra sujeto" (*within subject*). Mientras que el grupo es un componente "entre sujeto" (*between subject*). Por lo tanto, se trata de un diseño **1W+1B**.

Las técnicas clásicas de la tabla ANOVA y su inferencia són válidas siempre y cuando se cumpla la **condición de esfericidad**: la variancia de la diferencia entre dos medidas es constante. Para comprobar la condición de esfericidad se puede aplicar el **test de Mauchly**.

Si no se cumple hay que corregir los grados de libertad de los términos "intra sujetos" de la tabla ANOVA y se recalculan sus p-valores. Hay dos métodos para **corregir los grados de libertad**: método "Huynh and Feldt" (H-F) y el método "Greenhouse-Geisser" (G-G) .

5.1.2 Diseño 1W

Si en el diseño no hay grupos, luego el modelo se simplifica a un diseño de un solo factor "intra sujeto" **(1W)**

$$y_{ij} = \mu + \pi_i + \beta_j + e_{ij}$$

En ambos casos, tanto en el diseño en que tenemos grupos (1W+1B) como en el que no (1W), no nos interesa evaluar el efecto del individuo; ya sabemos que hay variabilidad entre ellos. Veremos en un ejemplo como el paquete **`ez`** que se usará para esta técnica de suma de cuadrados omite los resultados sobre el factor aleatorio individuo.

5.1.3 Función `ezANOVA`

Para ajustar los modelos de medidas repetidas balanceados mediante la técnica de suma de cuadrados existe la función **`ezANOVA`** del paquete ez.

```
library(ez)
```

Tanto la corrección por H-F o G-G, como el test de esfericidad de Mauchly estan implementados en el package `ez` de R. Para visualizar gráficamente los

resultados, se usará la función `ezPlot()`. Más adelante en esta sección se verá en un ejemplo de ambas funciones. Para llevar a cabo los análisis ANOVA se usa la función `ezANOVA ()` que tiene los siguientes argumentos:

- `data`: base de datos donde se encuentran las variables
- `dv`: variable respuesta o variable dependiente
- `wid`: variable individuo
- `within`: factor o factores "intra sujeto". Típicamente en este argumento se espedificará el tiempo. Si se especifica más de un factor, éstos deben estar cruzados y se escribirá `.(var1,var2)`.
- `between`: factor o factores "entre sujetos". Si no hay ningún factor "intra-sujeto" se deja a `NULL`. Como en el argumento `within`, si hay más de un factor "entre sujetos", éstos deben estar cruzados y se escribirá `.(var1,var2)`.

Observaciones:

- Los datos deben estar en formato vertical.
- La variable respuesta y los factores deben escribirse sin comillas.
- Los factores "intra", "entre" y el sujeto deben estar en format `factor`.
- El factor individuo debe tener tantos niveles como individuos.
- Aunque en teoría la función permite covariables (variables independientes contínuas), esta opción está en versión "beta".
- Todos los factores, excepto el individuo, deben ser de efectos fijos.

5.2 Respuesta Multivariante

Esta metodología también conocida como MANOVA asume que las observaciones de cada individuo es un vector multivariante donde la variable respuesta se considera dicho vector. Podemos escribirlo de la siguiente forma:

$$\vec{y}_i = \vec{\mu}_i + \vec{e}_i$$

Donde

- $\vec{y}_i = (y_{i1}, \dots, y_{iT})$ es el vector de medidas para el individuo i.
- $\vec{\mu}_i = (\mu_{i1}, \dots, \mu_{iT})$ es el vector con las medias de cada momento y para cada individuo. Las medias pueden depender de las variables independientes x_k. Fíjate que el coeficiente β_{kj} puede ser diferente para cada momento.

$$\mu_{ij} = \sum_{k=1}^{K} \beta_{jk} x_{ik}, \quad j = 1, ..., T$$

- $\vec{e_i} \sim N(\vec{0}, \Sigma)$, donde Σ es la matrix de covarianzas de los errores y tiene que ser la misma para todos los individuos. Su estructura, pero, puede ser cualquiera.
- x_{ik} valor de la variable independiente k del individuo i.

Observaciones

- Para ajustar este modelo los datos se disponen de forma horizontal (ancho).
- En este modelo los tiempos en que se toman las T medidas tienen que ser los mismos para todos los individuos.
- Para estudiar la evolución en el tiempo se puede realizar un contraste polinómico en el vector de medias $\vec{\mu}$.
- Para comparar grupos de medidas, por ejemplo si se tienen cinco medidas, las dos primeras corresponden al tratamiento A y las otras tres al tratamiento B, se puede realizar un contraste lineal para comparar los dos tratamientos.
- Cuando hay un valor faltante en alguna medida, toda la fila del individuo se tiene que eliminar.
- Cada variable independiente, x_{ki} es un único valor por individuo. O sea, que este modelo no contempla que las variables independientes sean de medidas repetidas. Si tuviéramos una variable que cambiara en el tiempo, se tienen que poner como variables diferentes (una para cada momento).
- Los factores contribuyen con tantas dummy variables como categorías menos uno en los términos x_{ik}.
- Los términos x_{ik} pueden ser también interacciones entre variables, como el producto de sus términos.

Datos

Matriz de diseño `~ fumador + edad + sexo + edad:fumador`

indiv

edad

fumador

sexo

fumadorEx

fumadorNunca

edad

sexomujer

fumadorEx:edad

fumadorNunca:edad

1

50

Ex

mujer

1

0

50

1

50

0

2

55

Actual

mujer

0

0

55

1

0

0

3

60

Actual

hombre

0

0

60

0

0

0

4

65

Nunca

mujer

0

1

65

1

0

65

5

62

Ex

hombre

1

0

62

0

62

0

5.3 Ejemplos

Vamos a ver algunos ejemplos que se analizarán mediante las técnicas que se acaban de describir.

5.3.1 Ejemplo 1

En la base de datos "Ejemplo_1W.csv" se tienen los datos de un diseño con 12 individuos en los que se toman los niveles en sangre de un cierto parámetro lipídico. Para cada invidivuo se miden los niveles a 1, 2 y 3 horas.

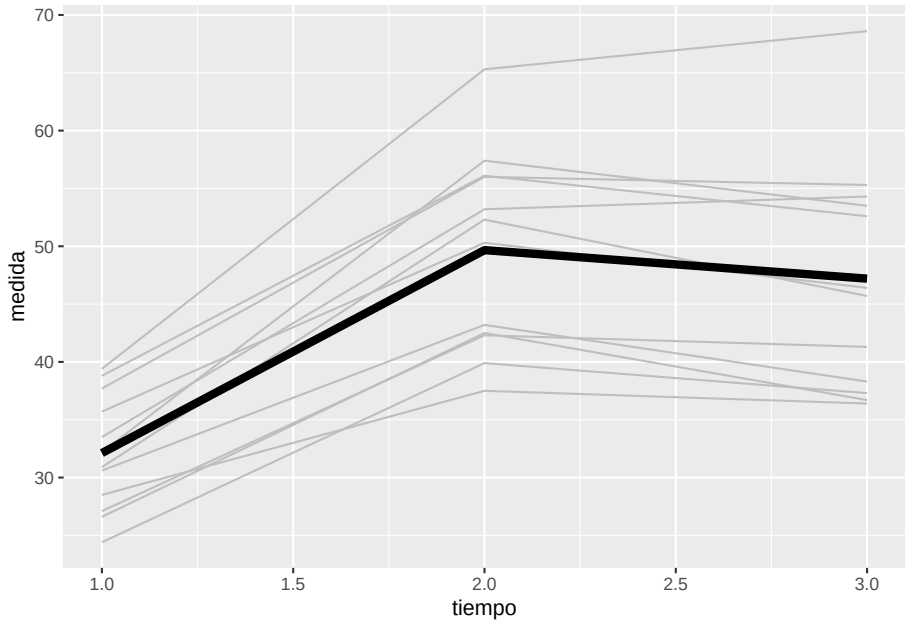
```
datos <- read.csv2("datos/Ejemplo_1W.csv")
```

Ordenamos por individuo y dentro por tiempo dentro de individuo

```
datos <- arrange(datos, indiv, tiempo)
```

5.3.1.1 Exploración de los datos

```
library(ggplot2)
p <- ggplot(data = datos, aes(x = tiempo, y = medida, group = indiv))
p + geom_line(col="grey") + stat_summary(aes(group = 1),
    geom = "line", fun = mean, size=2)
```



Cada línea representa a un individuo, mientras que la línea más gruesa es el promedio de los 12 individuos. Vemos como el efecto del tiempo no es del todo lineal. Además las líneas están bastante separadas indicando variabilidad entre los individuos.

Comprobemos si tenemos algún inviduo con datos faltantes

```
sum(with(datos, tapply(is.na(medida), indiv, any)))
```

```
[1] 0
```

Como son datos balanceados, podemos usar ANOVA y MANOVA

5.3.1.2 Suma de cuadrados (ANOVA)

Para ajustar este modelo hay que usar los datos en disposición vertical. Además, hay que convertir las variables `tiempo` e `indiv` a factor.

```
library(ez)

datos.ez <- datos
datos.ez$tiempo <- factor(datos.ez$tiempo)
datos.ez$indiv <- factor(datos.ez$indiv)

ezANOVA(data=datos.ez, dv=medida, wid=indiv, within=tiempo, detailed = TRUE)
```

```
$ANOVA
       Effect DFn DFd       SSn       SSd
1 (Intercept)   1  11 66546.801 1892.0589
2      tiempo   2  22  2166.376  264.6244
          F            p p<.05       ges
1 386.88796 6.390053e-10     * 0.9686088
2  90.05264 2.542699e-11     * 0.5011210

$`Mauchly's Test for Sphericity`
  Effect         W          p p<.05
2 tiempo 0.4433135 0.01712201     *

$`Sphericity Corrections`
  Effect       GGe        p[GG] p[GG]<.05
2 tiempo 0.6423901 5.662497e-08         *
       HFe        p[HF] p[HF]<.05
2 0.6905331 1.998401e-08         *
```

La condición de esfericidad no se cumple dado que el test de Mauchly es significativo. Por lo tanto, hay que corregir los grados de libertad y, en consecuencia, el p-valor del factor tiempo. Después de la corrección, éste sigue siendo significativo.

5.3.1.3 Modelo de respuesta multivariante (MANOVA)

Para analizar los datos mediante el modelo de respuesta multivariante hay que disponer los datos de forma horizontal.

```
datosh <- dcast(datos, indiv ~ paste0("medida_", tiempo),
                value.var = "medida" )
datosh
```

```
   indiv medida_1 medida_2 medida_3
1      1     39.4     65.3     68.6
2      2     33.5     53.2     54.3
3      3     27.1     42.3     41.3
4      4     30.9     52.3     45.7
5      5     32.2     57.4     53.5
6      6     26.6     42.5     36.7
```

```
7      7     28.5     37.5     36.4
8      8     37.7     56.0     55.3
9      9     35.7     50.3     46.4
10    10     30.6     43.2     38.3
11    11     24.4     39.9     37.3
12    12     38.8     56.1     52.6
```

Para ajustar un modelo de regresión lineal con respuesta multivariante se puede usar la función `lm`. Y hay que poner la variable respuesta a la izquierda de `~` como una matriz de las tres variables (`medida.1`, `medida.2` y `medida.3`):

```
respuesta <- as.matrix(datosh[,c("medida_1","medida_2","medida_3")])
modelo <- lm(respuesta ~ 1, data=datosh)
class(modelo)
```

```
[1] "mlm" "lm"
```

```
summary(modelo)
```

```

Response medida_1 :

Call:
lm(formula = medida_1 ~ 1, data = datosh)

Residuals:
    Min      1Q  Median      3Q     Max
-7.7167 -3.9667 -0.5667  4.0833  7.2833

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   32.117      1.445   22.23 1.72e-10

(Intercept) ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.006 on 11 degrees of freedom


Response medida_2 :

Call:
lm(formula = medida_2 ~ 1, data = datosh)

Residuals:
    Min      1Q  Median      3Q     Max
-12.167  -7.217   1.633   6.358  15.633
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   49.667      2.456   20.22 4.75e-10

(Intercept) ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.509 on 11 degrees of freedom


Response medida_3 :

Call:
lm(formula = medida_3 ~ 1, data = datosh)

Residuals:
   Min     1Q Median     3Q    Max
-10.80  -9.15  -1.15   6.50  21.40

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   47.200      2.867   16.47 4.25e-09

(Intercept) ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.93 on 11 degrees of freedom
```

Para obtener la matriz de covarianzas de los residuos:

```
estVar(modelo)
```

```
         medida_1 medida_2 medida_3
medida_1 25.05970 36.58970 42.08727
medida_2 36.58970 72.39879 81.30000
medida_3 42.08727 81.30000 98.60364
```

Y a partir de la matriz de covarianzas, se puede calcular fácilmente la matriz de correlaciones de los residuos:

```
cov2cor(estVar(modelo))
```

```
          medida_1  medida_2  medida_3
```

```
medida_1 1.0000000 0.8590239 0.8466744
medida_2 0.8590239 1.0000000 0.9622290
medida_3 0.8466744 0.9622290 1.0000000
```

Para obtener los resultados se usa la función `anova` (`?anova.mlm`)

```
anova(modelo, X = ~1, test = "Pillai")
```

```
Analysis of Variance Table


Contrasts orthogonal to
~1

            Df Pillai approx F num Df den Df
(Intercept)  1  0.945   85.903      2     10
Residuals   11
               Pr(>F)
(Intercept) 5.035e-07 ***
Residuals
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Los estadísticos disponibles (argumento `test`) son: "Pillai", "Wilks", "Hotelling-Lawley", "Roy" o "Spherical".

Con la opcion `X=~1` se contrasta si $\mu_1 = \mu_2 = \mu_3$. En cambio la opción por defecto `X = ~ 0`, contrasta $\mu_1 = \mu_2 = \mu_3 = 0$ que no es de interés.

El término `(Intercept)` corresponde al efecto del tiempo.

Resultado

Hay efecto del tiempo porque el p-valor < 0.05.

5.3.2 Ejemplo 2

En la base de datos "Ejemplo_1W1B.csv" se tienen los datos de un estudio en el que participan 24 individuos randomizados en dos grupos de tratamiento (`trat`). Como en el anterior ejemplo, para cada invidivuo se miden los niveles a 1, 2 y 3 horas.

```
datos <- read.csv2("datos/Ejemplo_1W1B.csv")
```

Como antes, ordenamos por individuo (de 1 a 24) y por tiempo

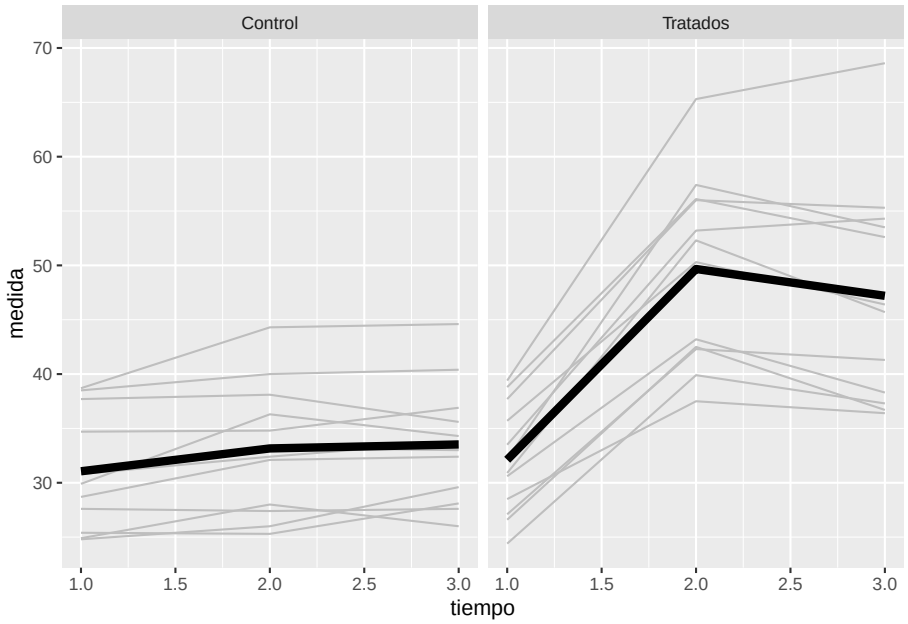
```
datos <- arrange(datos, indiv2, tiempo)
```

Fíjate que hay dos variables que codifican al individuo: la variable `indiv` va de 1 a 12 que son los individuos que hay dentro de cada grupo de tratamiento,

mientras que `indiv2` va de 1 a 24 que son el total de individuos.

5.3.2.1 Exploración de los datos

```
datos$trat <- factor(datos$trat, 1:2, c("Control","Tratados"))

library(ggplot2)
p <- ggplot(data = datos, aes(x = tiempo, y = medida, group = indiv2))
p <- p + geom_line(col="grey") + stat_summary(aes(group = 1),
    geom = "line", fun = mean, size=2)
p + facet_grid( ~ trat)
```



Para `trat=1`, la medida parece que no sube o sube muy poco. Mientras que para `trat=2` sube mucho hasta la segunda medida y se estabiliza en la tercera medida. Por lo tanto, parece que sí hay una interacción entre el tiempo y el grupo de tratamiento.

5.3.2.2 Suma de cuadrados (ANOVA)

Para ajustar este modelo hay que usar los datos en disposición vertical. Como antes hay que convertir las variables `tiempo`, `indiv2` y `trat` a factor.

```
library(ez)

datos.ez <- datos
datos.ez$tiempo <- factor(datos.ez$tiempo)
```

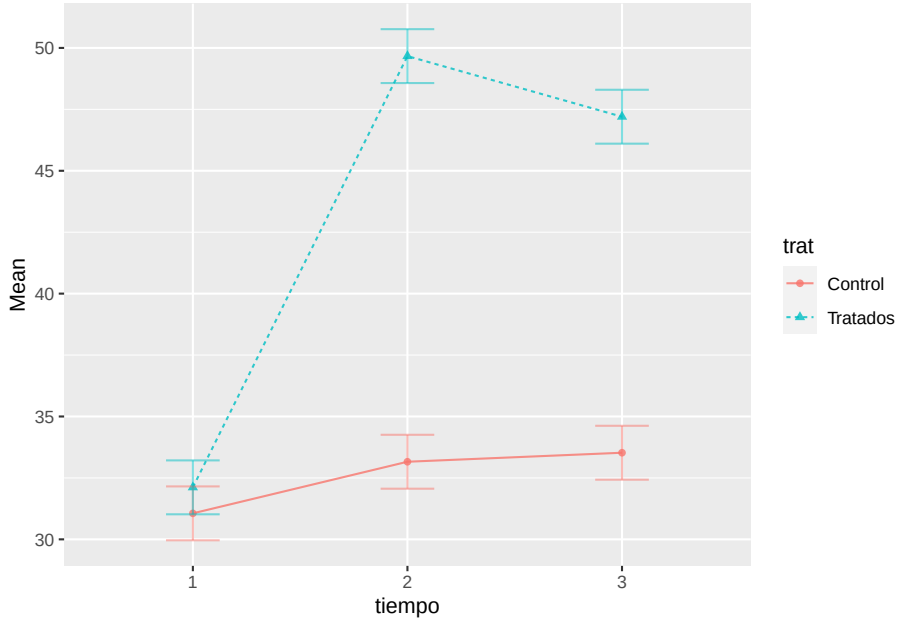
```
datos.ez$indiv2 <- factor(datos.ez$indiv2)
datos.ez$trat <- factor(datos.ez$trat)

ezANOVA(data=datos.ez,
        dv=medida,
        wid=indiv,
        within=tiempo,
        between=trat,
        detailed = TRUE)
```

```
$ANOVA
       Effect DFn DFd         SSn       SSd
1 (Intercept)   1  22 102808.4513 2849.9219
2        trat   1  22   1952.0835 2849.9219
3      tiempo   2  44   1397.0700  312.7422
4 trat:tiempo   2  44    811.8211  312.7422
          F            p p<.05       ges
1 793.63083 9.363797e-19     * 0.9701554
2  15.06913 8.040878e-04     * 0.3816578
3  98.27755 5.878117e-17     * 0.3063929
4  57.10794 5.921847e-13     * 0.2042582

$`Mauchly's Test for Sphericity`
       Effect         W           p p<.05
3      tiempo 0.5725954 0.002866835     *
4 trat:tiempo 0.5725954 0.002866835     *

$`Sphericity Corrections`
       Effect       GGe        p[GG] p[GG]<.05
3      tiempo 0.7005722 1.520842e-12         *
4 trat:tiempo 0.7005722 1.003472e-09         *
        HFe        p[HF] p[HF]<.05
3 0.7336894 4.932994e-13         *
4 0.7336894 4.400536e-10         *
```

Vemos como se aplican las correcciones sólo en los términos "intra sujeto" que son `tiempo` y la interacción `trat:tiempo`, ya que el test de Mauchly es significativo (p-valor < 0.05). Una vez aplicados las correcciones sobre los grados de libertad, los p-valores cambian aunque las conclusiones son las mismas: tanto el efecto del tiempo como su interacción con el tratamiento son significativos.

```
ezPlot(data=datos.ez,
       dv=medida,
       wid=indiv,
       within=tiempo,
       between=trat,
```

```
        x=tiempo,
        split=trat)
```



Las conclusiones con la tabla ANOVA corregida (tanto por GG como por HF), se ven claramente en el gráfico de interacción.

5.3.2.3 Modelo de respuesta multivariante (MANOVA)

Para analizar los datos mediante el modelo de respuesta multivariante, como antes hay que disponer los datos de forma horizontal.

```
datosh <- dcast(datos, indiv + trat ~ paste0("medida_", tiempo),
                value.var = "medida" )
datosh
```

```
   indiv     trat medida_1 medida_2 medida_3
1      1  Control     34.7     34.8     36.9
2      1 Tratados     39.4     65.3     68.6
3      2  Control     38.7     44.3     44.6
4      2 Tratados     33.5     53.2     54.3
5      3  Control     28.7     32.1     32.4
6      3 Tratados     27.1     42.3     41.3
7      4  Control     30.8     32.4     33.8
8      4 Tratados     30.9     52.3     45.7
9      5  Control     29.9     36.3     34.3
10     5 Tratados     32.2     57.4     53.5
```

```
11     6  Control     27.6     27.4     27.6
12     6 Tratados     26.6     42.5     36.7
13     7  Control     24.9     28.0     26.0
14     7 Tratados     28.5     37.5     36.4
15     8  Control     37.7     38.1     35.6
16     8 Tratados     37.7     56.0     55.3
17     9  Control     31.0     33.2     33.0
18     9 Tratados     35.7     50.3     46.4
19    10  Control     25.4     25.3     28.1
20    10 Tratados     30.6     43.2     38.3
21    11  Control     24.8     26.0     29.6
22    11 Tratados     24.4     39.9     37.3
23    12  Control     38.5     40.0     40.4
24    12 Tratados     38.8     56.1     52.6
```

```
respuesta <- as.matrix(datosh[,c("medida_1","medida_2","medida_3")])
modelo <- lm(respuesta ~ trat, data=datosh)
summary(modelo)
```

```
Response medida_1 :

Call:
lm(formula = medida_1 ~ trat, data = datosh)

Residuals:
    Min      1Q  Median      3Q     Max
-7.7167 -3.9667 -0.7083  4.1271  7.6417

Coefficients:
             Estimate Std. Error t value
(Intercept)    31.058      1.476  21.048
tratTratados    1.058      2.087   0.507
             Pr(>|t|)
(Intercept)  4.57e-16 ***
tratTratados    0.617
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.112 on 22 degrees of freedom
Multiple R-squared:  0.01156,   Adjusted R-squared:  -0.03337
F-statistic: 0.2572 on 1 and 22 DF,  p-value: 0.6171


Response medida_2 :
```

```
Call:
lm(formula = medida_2 ~ trat, data = datosh)

Residuals:
     Min       1Q   Median       3Q      Max
-12.1667  -6.6396   0.3375   5.2896  15.6333

Coefficients:
             Estimate Std. Error t value
(Intercept)    33.158      2.113  15.692
tratTratados   16.508      2.988   5.524
             Pr(>|t|)
(Intercept)  1.98e-13 ***
tratTratados 1.50e-05 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.32 on 22 degrees of freedom
Multiple R-squared:  0.5811,    Adjusted R-squared:  0.5621
F-statistic: 30.52 on 1 and 22 DF,  p-value: 1.495e-05


Response medida_3 :

Call:
lm(formula = medida_3 ~ trat, data = datosh)

Residuals:
     Min       1Q   Median       3Q      Max
-10.8000  -5.9063  -0.6625   5.6250  21.4000

Coefficients:
             Estimate Std. Error t value
(Intercept)    33.525      2.310  14.511
tratTratados   13.675      3.267   4.186
             Pr(>|t|)
(Intercept)  9.55e-13 ***
tratTratados 0.000384 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.003 on 22 degrees of freedom
Multiple R-squared:  0.4433,    Adjusted R-squared:  0.418
F-statistic: 17.52 on 1 and 22 DF,  p-value: 0.0003835
```

```
anova(modelo, X=~1)
```

```
Analysis of Variance Table


Contrasts orthogonal to
~1

            Df  Pillai approx F num Df den Df
(Intercept)  1 0.88537   81.102      2     21
trat         1 0.83628   53.633      2     21
Residuals   22
               Pr(>F)
(Intercept) 1.326e-10 ***
trat        5.599e-09 ***
Residuals
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Con la instrucción `summary`, contrasta si la media es diferente entre los dos grupos de tratamiento, y esto lo hace para cada momento por separado.

En la tabla ANOVA, el p-valor de término `trat` contrasta si el efecto del tiempo es el mismo para los dos tratamientos, o sea, la interacción tratamiento y tiempo, que es lo que nos interesa. Mientras que el término `(Intercept)` corresponde al efecto marginal del tiempo.

Fíjate qué pasa si no se especifica el argumento `X`:

```
anova(modelo)
```

```
Analysis of Variance Table

            Df  Pillai approx F num Df den Df
(Intercept)  1 0.97752  289.829      3     20
trat         1 0.84439   36.176      3     20
Residuals   22
               Pr(>F)
(Intercept) < 2.2e-16 ***
trat        2.854e-08 ***
Residuals
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En este caso, el p-valor del tratamiento contrasta si hay diferencias entre tratamientos en alguno de los momentos (hay 3 grados de libertad). Y este

contraste no es la interacción entre tratamiento y tiempo.

5.3.3 Ejemplo 3

En un estudio se quieren comparar el efecto de régimen de ejercicio sobre el sobrepeso. Para ello se reclutan 100 personas. A la mitad se le asigna el régimen y al resto se le hacen algunas recomendaciones (grupo control). Se mide el índice de masa corporal justo antes de empezar el estudio (momento basal), y al cabo de 1, 2 y 3 semanas. Como la edad es una variable importante para predecir el IMC también se registra.

Los datos los encontrarás en el fichero "imc.csv"

En este ejemplo, vemos como en algunos de los individuos nos falta alguna medida en a partir de la primera semana. Por este motivo usaremos la técnica de los LMM.

```
datos <- read.csv2("datos/imc.csv")
```

Nos aseguramos que los datos estén ordenados por individuo y tiempo

```
datos <- arrange(datos, indiv, tiempo)
```

Recodificamos nuestra variable tratamiento:

```
datos$tx <- factor(datos$tx, 1:2, c("Control", "Tratados"))
summary(datos)
```

```
   respuesta         indiv            tiempo
 Min.   : 9.80   Min.   :  1.00   Min.   :0.00
 1st Qu.:27.02   1st Qu.: 25.75   1st Qu.:0.75
 Median :30.75   Median : 50.50   Median :1.50
 Mean   :30.46   Mean   : 50.50   Mean   :1.50
 3rd Qu.:34.60   3rd Qu.: 75.25   3rd Qu.:2.25
 Max.   :43.70   Max.   :100.00   Max.   :3.00
 NA's   :50
      edad             tx
 Min.   :25.00   Control :200
 1st Qu.:43.00   Tratados:200
 Median :49.00
 Mean   :49.03
 3rd Qu.:57.00
 Max.   :69.00
```

Comprobemos si tenemos algún inviduo con datos faltantes

```
sum(with(datos, tapply(is.na(respuesta), indiv, any)))
```

```
[1] 42
```

Como hay individuos con datos faltantes no podemos utilizar ANOVA o MANOVA y debemos usar modelos más avanzados que veremos más adelante.

5.4 Ejercicios

5.4.1 Ejercicio 1

Para estudiar las diferencias entre dos procedimientos diferentes de recuperación de pacientes de un infarto, se consideraron dos grupos experimentales en sendos hospitales, de 8 pacientes cada uno. La variable respuesta es el índice de Bartel, que varía entre 0 y 100, y que constituye una medida de la habilidad funcional con la que se valoran diferentes capacidades, de forma que valores más altos se corresponden con una mejor situación del paciente. De cada uno de los 16 pacientes se dispone de su respuesta cada semana a lo largo de 5 semanas consecutivas. Los datos se pueden encontrar en el archivo *recuperainfarto.txt*. El fichero contiene la información para cada individuo en una fila, la primera columna contiene la información del hostpital y las siguiente 5 columnas corresponden al valor del índice para cada semana.

1. Carga los datos en R.
2. Nombra a las columnas del data.frame con 'c("hospital", "bartel_1", "bartel_2", "bartel_3"), "bartel_4", "bartel_5")
3. Añade una columna (`id`) a los datos que corresponda al identificador cada individuo (usa 1, 2, 3, … 16).
4. Crea un data.frame con los datos en formato largo
5. Crea una figura para visualizar la evolución en la respuesta a lo largo del tiempo para cada individuo.
6. Crea una figura para mostrar si visualmente hay diferencias entre ambos procedimientos a lo largo del tiempo.
7. ¿Qué procedimiento presenta una mejor recuperación de los pacientes? ¿Es esta diferencia estadísticamente significativa?

5.4.2 Ejercicio 2

En un estudio sobre la agudeza visual se dispone de la respuesta de siete individuos. La respuesta en cada ojo es el retraso en milisegundos entre la emisión de una luz y la respuesta en a la misma por el cortex. Cada ojo se somete a cuatro mediciones correspondientes a cuatro instantes consecutivos. Se tienen mediciones en el ojo izquierdo y derecho. Los datos se pueden encontrar en el archivo *agudezavisual.txt*

1. Crea una nueva base de datos agregando la información para cada una de la medida repetida (ojo) [NOTA: toma la media -

usa la función `aggregate` o cualquier otra que creas oportuna]. Usando esta nueva base de datos, contesta a las siguientes preguntas:

2. Crea una figura para mostrar si visualmente hay diferencias en el retraso promedio de ambos ojos a lo largo del tiempo para cada individuo.
3. ¿Existen diferencias entre la medición final y la basal?
4. ¿Existe un efecto temporal en la respuesta?

5.4.3 Ejercicio 3

Los datos `o2cons`, disponibles en el paquete `MANOVA.RM`, contiene medidas sobre el consumo de oxígeno de los leucocitos ("O2") de 144 individuos, 72 de ellos asignados al grupo control ("Group=P") y el resto al grupo de tratamiento con Verum (Group=V). Además, para cada individuo se recoge si los estafilococos ("Staphylococci") estaban activados o no (0/1). Para cada individuo se tomaron los niveles de oxígeno de los leucocitos después de 6, 12 y 18 minutos.

1. Crea otro data.frame con los datos en formato ancho
2. Haz una pequeña descriptiva de los datos contenidos en esta base de datos
3. Analiza la evolución del consumo de oxígeno del grupo de tratamiento ("Group=V").
4. Crea una figura para mostrar si visualmente hay diferencias en la evolución del comsumo de O2 entre el grupo de intervención y el grupo de tratamiento.
5. ¿Son estas diferencias estadísticamente significativas?

Recuerda que los datos los puedes cargar mediante la instrucción

```
library(MANOVA.RM)
data(o2cons)
```

Chapter 6

Modelos Lineales Mixtos (LMM)

Esta técnica es la más potente para analizar datos longitudinales ya que permite introducir efectos aleatorios y especificar la estructura de correlaciones de los residuos dentro de un mismo individuo.

Además, a diferencia de las dos técnicas anteriores, permite trabajar con missings.

6.1 Ecuación

$$y_{ij} = \beta_{0i} + \sum_{k=1}^{K} \beta_{ki} x_{ijk} + e_{ij}$$

Donde i representa al individuo, j representa el momento (de uno hasta hasta el número de observaciones del individuo i),

- x_{ijk} valor de la k-ésima variable independiente del individuo i en el momento j.
- $\beta_{0i} \sim N\left(\beta_0, \sigma^2_{\beta_0}\right)$ es la constante del modelo aunque en general se supone aleatoria, o sea que tiene cierta varianza entre individuos y está centrada en la contaste μ.
- $\beta_{ki} \sim N\left(\beta_k, \sigma^2_{\beta_k}\right)$: pendientes o coeficientes de las variables del modelo. Pueden ser aleatorias, o sea, variar entre individuos.

En general puede haber correlación entre la contante β_{0i} y las pendientes β_{ki}.

El vector formado por la constante y por los coeficientes aleatorios, son los **efectos aleatorios** y se supone que sigue una distribución normal multivariada:

$$\vec{\beta}_i = (\beta_{0i}, \beta_{1i}, \dots, \beta_{Ki})^t \sim N\left(\vec{\beta}, \Omega\right)\Big)$$

- El vector formado por los errores de un individuo $\vec{e_i} \sim N(\vec{0}, \Sigma_i)$, sigue una distribuión normal multivariante con una cierta matriz de covarianzas Σ_i que no tiene porqué ser la misma ni del mismo tamaño para todos los individuos ya que no todos los individuos tendrán el mismo número de observaciones. **Los errores son independientes de la constante aleatoria y de los coeficientes aleatorios**.

Observaciones

- Para ajustar este modelo los datos se disponen de forma vertical.
- El modelo LMM es muy flexible y potente. No sólo permite especificar efectos aleatorios con lo que evaluar la variabilidad de ciertos efectos o variables entre individuos sinó también la correlación residual entre las distintas medidas repetidas en un mismo individuo.
- Cuando hay missings en una observación no hace falta eliminar las otras del mismo individuo, ya que cada fila aquí es una observación y no un individuo.
- La esperanza de la constante y coeficientes aleatorios $\vec{\beta}_i$ es la misma para todos los individuos, $\vec{\beta}$, y la matriz de covarianzas, Ω, también (**homocedesticidad**).
- Si un coeficiente no es aleatorio, se puede notar como $\beta_{k'i} = \beta_{k'}$ en lugar de suponer que sigue una distribución normal. También se podría pensar que sigue una distribución "normal" con varianza cero.
- Los **efectos fijos** son la esperanza de los efectos aleatorios $(\beta_0, \beta_1, \dots, \dots, \beta_k)$. Además, cuando un coeficiente no es aleatorio (tiene varianza cero) se denomina fijo directamente.
- Hay un número limitado de efectos aleatorios que se pueden incorporar en el modelo, que no puede exceder el número de medidas por individuo.
- La presencia de **efectos aleatorios inducen correlación** entre medidas de un mismo individuo. Sin embargo, según que estructura de correlación sólo se puede conseguir definiendo también una estructura de correlación entre residuos no nula (no diagonal).
  - Considerando los coeficientes del tiempo como aleatorios se induce correlación distinta según los tiempos que se toman las medidas.
  - Considerando el coeficiente de una variable **no** cambiante en el tiempo se induce **heterocedesticidad** (varianza diferente) entre los individuos.

Por ejemplo, supongamos un modelo con efecto lineal del tiempo y una covariable no cambiante del tiempo (x_i). Y tomamos la constante y el coeficiente de x_i aleatorios y ambos no correlacionados.

$$y_{ij} = \beta_{0i} + \beta_1 t_{ij} + \beta_{2i} x_i$$

Luego la varianza de y_{ij} es $\sigma^2_{\beta_0} + \sigma^2_{2i} x_i^2$. Si x_i vale 0 para el grupo placebo y 1 para los tratados, entonces la varianza del grupo placebo será $\sigma^2_{\beta_0}$ y para los tratados $\sigma^2_{\beta_0} + \sigma^2_{\beta_2}$

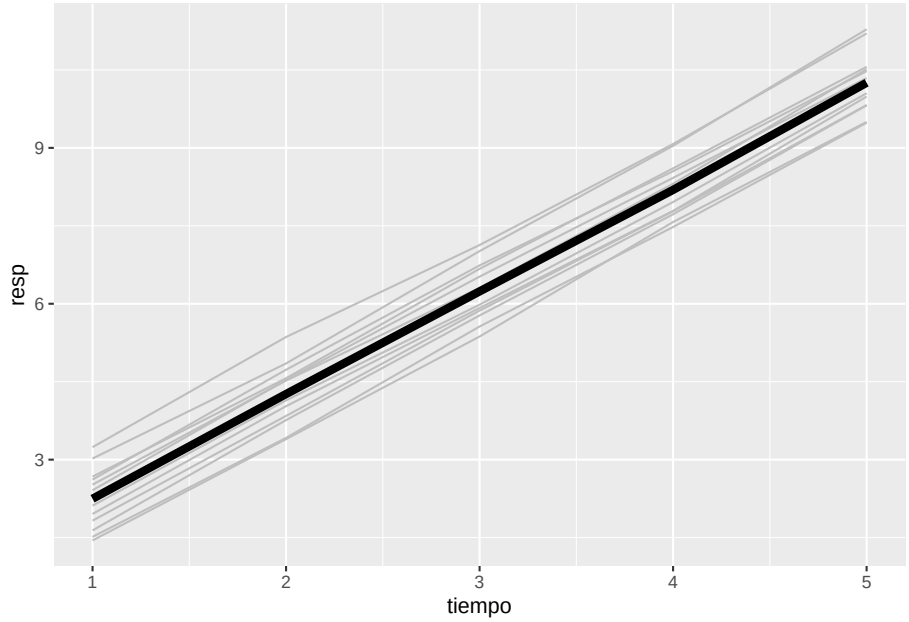
- A diferencia de las técnicas de sumas de cuadrados y de respuesta multivariante, en que la variable **tiempo** se trata como a un **factor**, con los LMM se tratar también como **variable continua**.

6.2 Casos particulares

6.2.1 Modelo con constante aleatoria

$$y_{ij} = \beta_{0i} + \beta_1 t_{ij} + e_{ij}$$

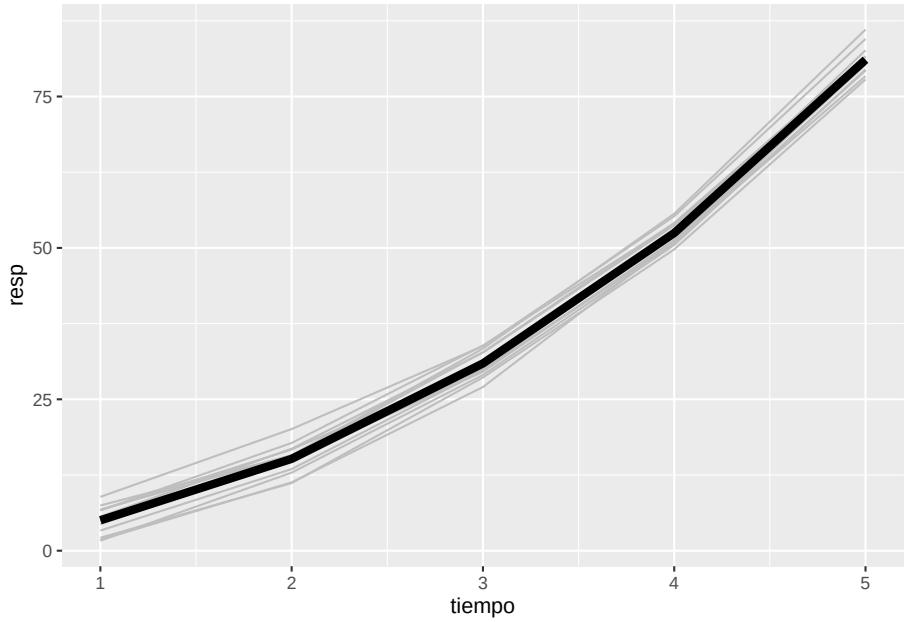
Donde $\beta_{0i} \sim N(\beta_0, \sigma_{\beta_0})$, y β_1 es el coeficiente fijo del tiempo. En este caso se supone que el tiempo tiene un efecto lineal.



Podríamos añadir un término cuadrático, cúbico, etc. si el efecto no fuera lineal y añadiéramos un término cuadrático:

$$y_{ij} = \beta_{0i} + \beta_1 t_{ij} + \beta_2 t_{ij}^2 + e_{ij}$$

en este caso, $x_{ij1} = t_{ij}$ y $x_{ij2} = t_{ij}^2$.



Correlacion entre observaciones

Si los errores son independientes, las observaciones de la variable respuesta de un mismo individuo están correlacionadas. Y esta correlación es constante: no depende de la distancia entre las medidas.

$$\text{corr}(y_{i1}, y_{i2}) = \text{corr}(y_{i1}, y_{i3}) = ... = \frac{\sigma^2_{\beta_0}}{\sigma^2_e}$$

A esta correlación también se la conoce como **coeficiente de correlación intraclase (ICC)**

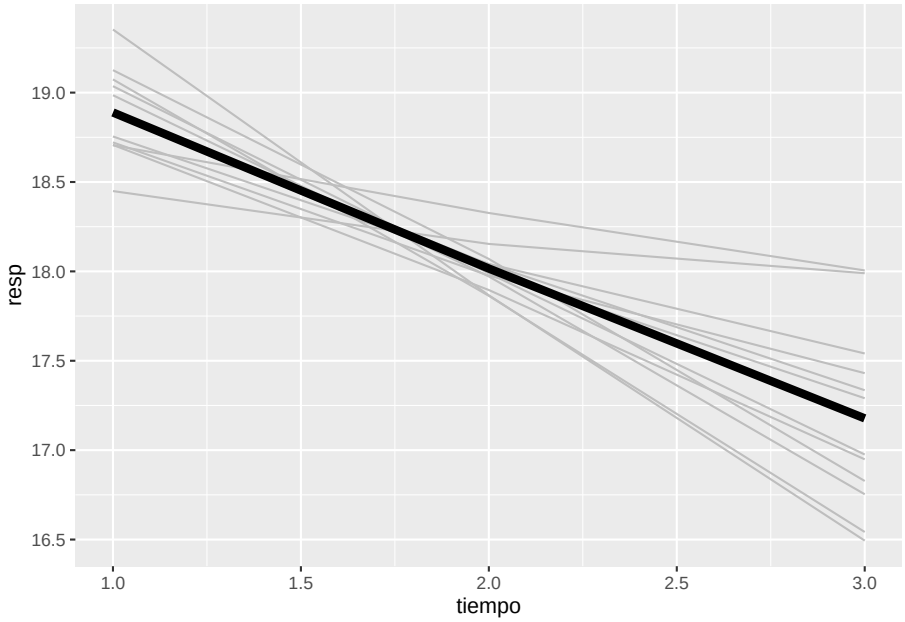
6.2.1.1 Modelo con pendiente y constante aleatoria

$$y_{ij} = \beta_{0i} + \beta_{1i} t_{ij} + e_{ij}$$

$\vec{\beta_i} = (\beta_{0i}, \beta_{1i})^t \sim N\left((\beta_0, \beta_1)^t, \Omega\right)$, donde

$$\Omega = \begin{pmatrix} \sigma^2_{\beta_0} & \sigma_{\beta_0,\beta_1} \\ \sigma_{\beta_0,\beta_1} & \sigma^2_{\beta_1} \end{pmatrix}$$

El término σ_{β_0,β_1} es la covarianza entre la constante y la pendiente. Ésta en general puede no ser cero.



En este gráfico se observa primero que las pendientes son diferentes entre los individuos. Y además, que los individuos que empiezan de más arriba bajan más rápido y viceversa. Así pues, la correlación entre la constante y la pendiente es negativa.

Correlacion entre observaciones

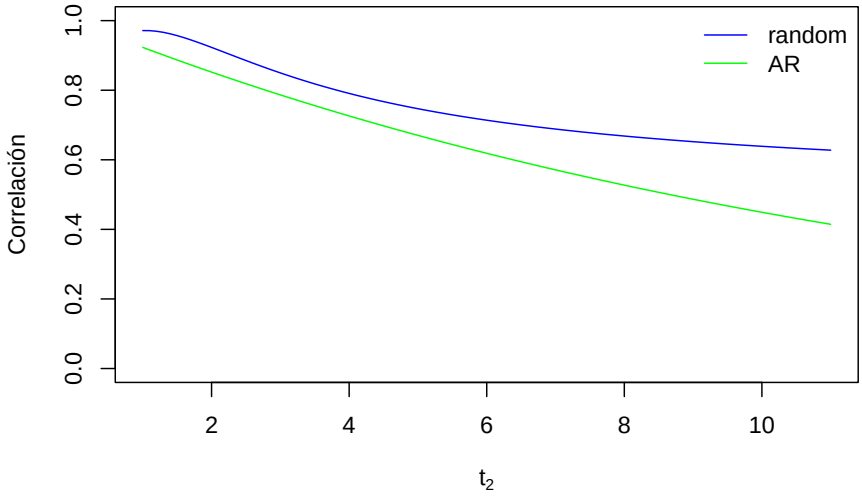
En el modelo con constante y pendientes aleatorios, si asumimos que los errores son independientes, las observaciones de la variable respuesta de un mismo individuo están correlacionadas. Y esta correlación depende de los momentos:

$$\text{corr}(y_{i1}, y_{i2}) = \frac{\sigma^2_{\beta_0} + \sigma_{\beta^2_{1i} \cdot t_1 \cdot t_2}}{\sqrt{\sigma^2_{\beta_0} + \sigma^2_{\beta_1} \cdot t_1^2 + \sigma^2_e}\sqrt{\sigma^2_{\beta_0} + \sigma^2_{\beta_1} \cdot t_2^2 + \sigma^2_e}}$$

Por lo tanto depende tanto de t_1 como de t_2 y no sólo de la distancia entre las medidas.

Si lo comparamos con el AR(1):

Ejemplo con $t_1 = 1$, $\sigma^2_{\beta_0} = 0.5^2$, $\sigma^2_{\beta_1} = 0.3^2$ y $\sigma^2_e = 0.1^2$, con $t_1 = 1$



En este ejemplo, vemos como especificando el AR y la pendiente fija, la correlación entre observaciones baja más rápidamente a medida que las observaciones se alejan (t_2) que lo que se consigue especificando la pendiente aleatoria y los errores incorrelacionados.

6.3 Simplificación del modelo

Empezaremos con el modelo más general, o sea, sin asumir independencia de los residuos, con efectos aleatorios (todos los que se admitan) correlacionados.

En cuanto a los efectos fijos, se incluyrán también los máximos que se puedan, interacciones si es pertinente, terminos cuadráticos (cúbicos), ...

A partir de aquí se simplificará el modelo en el siguiente orden:

6.3.1 Significación de los efectos aleatorios

La hipótesis nula para contrastar los factores de efectos aleatorios es que su varianza es igual a cero. Por ejemplo para la constante aleatoria:

$$\begin{cases} \mathrm{H}_0 : \sigma^2_{\beta_0} = 0 \\ \mathrm{H}_1 : \sigma^2_{\beta_0} > 0 \end{cases}$$

Hay diferentes técnicas estadísticas para contrastar estos tests, pero no son estándard. El problema es que la varianza de una distribución normal no puede

ser cero, por lo tanto la hipótesis nula está fuera del espacio parametrico ("beyond boundary"). Existen, pero, algunas herramientas en R que lo realizan mediante técnicas de remuestreo ("bootstrap"). Éstas son complejas desde el punto de visto teórico y no se explicarán en este curso (véase el paquete de R `pbkrtest` **?**). Otra alternativa es usar índices como el AIC o BIC (cuanto más bajo mejor), que proporciona la función `anova` en la comparación de dos modelos: uno considerando el coeficiente como aleatorio (β_{ik}) el otro considerando el coeficiente como fijo (β_k).

6.3.2 Elección matriz covarianzas de los efectos aleatorios

Si en el paso anterior, hay más de un efecto aleatorio significativo, seguidamente hay que contrastar si la correlación entre ellos es cero o no. Es decir, H_0 postula que la matriz Ω es diagonal, mientras que la H_1 se asume que las correlaciones pueden ser no nulas.

$$\begin{cases} H_0 : \Omega = \begin{pmatrix} \sigma^2_{\beta_0} & 0 \\ 0 & \sigma^2_{\beta_1} \end{pmatrix} \\ H_1 : \Omega = \begin{pmatrix} \sigma^2_{\beta_0} & \sigma_{\beta_0\beta_1} \\ \sigma_{\beta_0\beta_1} & \sigma^2_{\beta_1} \end{pmatrix} \end{cases}$$

Como la matriz diagonal es un caso particular de la matriz general, en que las correlaciones son cero se puede aplicar el test de razón de verosimilitudes.

6.3.3 Estructura de correlación de los errores

Mediante el **test de razón de verosimilitudes (LRT)**, se comparan las verosimilitudes de dos modelos.

Hay que ajustar el modelo mediante el criterio de máxima verosimilutud.

Los modelos tienen que estar anidados: la matriz de covarianzas de los errores de un modelo se pueda expresar como un caso particular de la del otro modelo. Por ejemplo, la matriz sin estructura sería la más general de todas, y la matriz de simetría compuesta sería un caso particular en que todas las correlación son iguales. No están anidadas las matrices con estructura MA(1) y una AR(1).

La simetría compuesta es un caso particular de matriz sin estructura.

$$\begin{pmatrix} 1 & \rho_{12} & \rho_{13} \\ \rho_{12} & 1 & \rho_{23} \\ \rho_{13} & \rho_{23} & 1 \end{pmatrix} \Rightarrow (\rho_{12} = \rho_{13} = \rho_{23} = \rho) \Rightarrow \begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix}$$

La matriz que supone independencia entre los residuos es un caso particular de matriz de simetría compuesta.

$$\begin{pmatrix} 1 & \rho & \rho \\ \rho & 1 & \rho \\ \rho & \rho & 1 \end{pmatrix} \Rightarrow (\rho = 0) \Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Pero no se puede pasar de una AR(1) a una MA(1) ni viceversa. En este caso el test LRT no es válido pero sí el criterio AIC o BIC.

$$\begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix} \Rightarrow (????) \Rightarrow \begin{pmatrix} 1 & \frac{\theta}{1+\theta^2} & 0 \\ \frac{\theta}{1+\theta^2} & 1 & \frac{\theta}{1+\theta^2} \\ 0 & \frac{\theta}{1+\theta^2} & 1 \end{pmatrix}$$

Una matriz AR(p') está anidada a AR(p) si p' < p, o sea un AR de orden menor está anidado a una de orden mayor, y por lo tanto, se puede aplicar un LRT para decidir el valor de p. Por ejemplo, un AR de orden 3 se especificaría como `correlation = corARMA(p=3, q=0)`. Lo mismo sucede para decidir el orden de una MA. Por ejemplo, para una MA de orden 4, `correlation = corARMA(p=0, q=4)`.

Heterocedesticidad:

La heterocedesticidad se produce cuando los parámetros de la matriz de covarianzas Σ dependen de variables. Por ejemplo, del sexo o de la edad, etc, o de una combinación lineal de las variables (valor esperado).

Por ejemplo, que la varianza sea distinta según el sexo, mientras que la correlación sea la misma:

para hombres

$$\Sigma_{\mathrm{H}} = \sigma^2_{\mathrm{H}} \begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix}$$

,

y para las mujeres

$$\Sigma_{\mathrm{M}} = \sigma^2_{\mathrm{M}} \begin{pmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{pmatrix}$$

También podríamos definir las varianzas (diagonal de Σ), en función del tiempo.

Veremos como es posible modelizar diferentes varianzas distintas entre grupos de individuos con la función `lme` de R que se describirá en esta sección.

6.3.4 Efectos fijos

Una vez escogida la estructura de covarianzas de los efectos aleatorios, de los errores, y qué efectos son aleatorios (contraste sobre sus varianzas), vamos a contrastar la significación de los efectos fijos:

Para ello, se puede usar el test de Wald para testar un único parámetro:

$$\begin{cases} \mathrm{H}_0 : \beta_1 = 0 \\ \mathrm{H}_1 : \beta_1 \neq 0 \end{cases}$$

o LRT para testar más de un parámetro a la vez, por ejemplo las dummies de un factor de más de dos categorías:

$$\begin{cases} \mathrm{H}_0 : \beta_1 = \beta_2 = 0 \\ \mathrm{H}_1 : \text{alguno diferente de } 0 \end{cases}$$

6.4 Validación del modelo

Una vez simplificado y seleccionado el modelo, hay que validarlo.

De todas las premisas a comprobar en este curso nos limitaremos a las asumciones sobre los **residuos**.

Para ellos se realizarán dos gráficos:

- **Residuos estandarizados vs valores predichos:** en este gráficos debería aparecer una nube de puntos uniformemente distribuida sin ninguna tendencia. Ésto nos indicaría que no nos hemos dejado ninguna variable, o ningún término cuadrático o cúbico del tiempo.
- **QQ-plot:** éste gráfico está pensado para comprobar la normalidad. Si los puntos se encuentran alrededor de la diagonal sin seguir ningún patrón, dará evidencia de que los residuos siguen una distribución normal

Hay otras premisas que se deberían comprobar, como por ejemplo la normalidad de los efectos aleatorios. Pero, por su complejidad, no se verá en este curso.

La valicación de los **efectos aleatorios** es más compleja. Una posibilidad "naive" es considerar que sus estimaciones siguen una distribución normal y que su distribución no depende de ninguna covariable a nivel de individuo. Veremos como las funciones de R para estimar los LMM proporcionan las estimaciones de los efectos aleatorios ("Empirical Bayes Estimates"). Aunque los efectos aleatorios se suponen normalmente distribuidos, los "Empirical Bayes Estimates" no tienen porqué.

6.5 Predicciones

6.5.1 Efectos marginales

Los efectos marginales representan el valor esperado de la variable respuesta. Para calcularlos hay que especificar los valores de las variables predictoras (condicionar):

$$E(Y_{ij}|x_{ij1}, \dots, x_{ijK}) = \beta_0 + \sum_{k=i}^{K} \beta_k x_{ijk}$$

Una vez ajustado el modelo con los datos de la muestra, se estiman los valores de los parámetros para estimar los efectos marginales o predicciones.

$$\hat{E}(Y_{ij}|x_{ij1}, \dots, x_{ijK}) = \hat{\beta}_0 + \sum_{k=i}^{K} \hat{\beta}_k x_{ijk}$$

6.5.2 Estimación de los efectos aleatorios

También podemos condicionar al individuo:

$$\hat{E}(Y_{ij}|x_{ij1}, \dots, x_{ijK}, i) = \hat{\beta}_{0i} + \sum_{k=i}^{K} \hat{\beta}_{ki} x_{ijk}$$

Donde $\hat{\beta}_{0i}$, $\hat{\beta}_{ki}$ son los "Empirical Bayes Estimates".

6.6 Función `lme`

Para ajustar los modelos lineales mixtos usaremos la función **`lme`** del paquete **`nlme`** [**?**]. Esta función permite incorporar efectos aleatorios, así como especificar la estructura de la matriz de correlaciones de los residuos.

El método se basa en el **criterio de máxima verosimilidud** ("Maximum Likelihood" - ML), que busca el valor de los parámetros que maximizan la función de verosimilitud. Generalmente, la solución no es una fórmula cerrada y se necesitan métodos iterativos numéricos para encontrar el óptimo. También se calculan mediante métodos numéricos la primera y segunda derivada para acelerar el proceso de estimación y para obtener los errores estándard de las estimaciones.

Para usar la función `lme`, los datos deben estar en formato horizontal. No hace falta que haya el mismo número de medidas para cada individuo, ni que las medidas se hayan producido en los mismos tiempos.

```
library(nlme)
```

```
?lme
```

Los **argumentos** más importantes de la función `lme`

- **`fixed`**: Fórmula de la forma

```
respuesta ~ var1 + var2 + var3
```

La constante se presupone que está y no hace falta escribir `1+`. La sintaxis es la misma que para el "formula environment" de otras funciones estándard como `lm` para regresión lineal ordinaria (los términos van separados con `+`, las interacciones se especifican con `:`, etc.). A la izquierda de `~` se especifica la variable respuesta.

- **`random`**: fórmula de la forma

```
 ~ var1 + var2 + ... + varK | indiv
```

sin ninguna variable a la izquierda de `~`, donde `indiv` es la variable sujeto y `var1`, `var2`, … `varK` son las variables con coeficiente aleatorio. Por defecto se supone que la constante está incluida.

Si se desea que la constante no sea aleatoria `~ var1 + ... + varK - 1 | indiv`.

Si sólo la constante es aleatoria `~ 1 | indiv`

Para especificar que la matriz Ω es diagonal se usa la función `pdDiag`

```
list(indiv = pdDiag( ~ var1 + var2 + ... + varK))
```

Si los individuos estuvieran anidados en clústers aleatorios: `~ var1+..| clusters / indiv`

- **correlation**: Para especificar la forma de la matriz de covarianzas de los residuos Σ_i. Para más estructuras: `?corClasses`
  - Residuos independientes (valor por defecto): `NULL`
  - Simetría compuesta: `corCompSymm()`
  - AR(1): `corAR1()`
  - ARMA(p,q): `corARMA(p,q)`
  - $\phi^{|t_i - t_j|}$: `corCAR1(form = ~ tiempo | indiv)`
  - Sin estructura | `corSymm()`

Para `corCAR1`, ϕ es la correlación entre dos medidas a distancia de una unidad de tiempo.

Importante!: para `corCompSymm`, `corAR1`, `corARMA` o `corSymm`, las medidas tienen que estar ordenadas dentro de cada individuo. Si no, hay que especificar la variable momento,

```
corAR1(form = ~ tiempo | indiv)
```

- **weights**: Este argumento modeliza la varianza, σ^2 según variables. Por defecto, `NULL` que supone que la matriz de covarianzas es la misma para todos los individuos. En lugar de una variable, puede ser el valor predicho, `varFixed(fitted(.))`. Para ver más `?varClasses`.
  - `varPower()`: $\sigma^2(x) = |x|^{2*\theta}$
  - `varFixed()`: $\sigma^2(x) = |x|$
  - `varConstPower()` $\sigma^2(x) = (\theta_1 + |x|^{\theta_2})^2$
- **method**: Método usado para estimar los parámetros (ML o REML). Para usar el LRT, o calcular los índices AIC o BIC se usa el método ML. La función `anova` que compara dos modelos por LRT, reajusta los modelos automàticamente bajo el método ML si han sido estimados con REML.
  - REML ("REstricted Maximum Likelihood"): método por defecto y que proporciona estimaciones no sesgadas de los parámetros.
  - ML ("Maximum Likelihood"): proporciona estimaciones de los parámetros sesgados.

6.7 Ejemplos

6.7.1 Ejemplo 1

Analicemos de nuevo el primer ejemplo que vimos en el anterior tema y que están disponibles en la base de datos "Ejemplo_1W.csv". Esta fichero contiene los datos de un diseño con 12 individuos en los que se toman los niveles en sangre de un cierto parámetro lipídico. Para cada invidivuo se miden los niveles a 1, 2 y 3 horas.

Recordemos que los datos se pueden cargar en R mediante

```
datos <- read.csv2("datos/Ejemplo_1W.csv")
```

Ordenamos por individuo y dentro por tiempo dentro de individuo

```
library(dplyr)
datos <- arrange(datos, indiv, tiempo)
```

Primero, ajustamos el modelo más complejo con constante y pendiente aleatoria, y añadimos el tiempo al cuadrado ya que vemos por el gráfico que la tendencia no es lineal.

```
modelo <- lme(fixed = medida ~ poly(tiempo, 2),
              data=datos,
              random = ~ poly(tiempo, 2) | indiv,
              correlation = corAR1(form = ~ tiempo | indiv)
              )
modelo$modelStruct$corStruct
```

```
Correlation structure of class corAR1 representing
         Phi
0.0001428372
```

```
summary(modelo)
```

```
Linear mixed-effects model fit by REML
  Data: datos
       AIC      BIC    logLik
  208.6888 225.1504 -93.34439

Random effects:
 Formula: ~poly(tiempo, 2) | indiv
 Structure: General positive-definite, Log-Cholesky parametrization
                 StdDev    Corr
(Intercept)       7.526056 (Intr) p(,2)1
poly(tiempo, 2)1 14.559990  0.881
poly(tiempo, 2)2  5.178443 -0.663 -0.729
Residual          1.441564

Correlation Structure: AR(1)
 Formula: ~tiempo | indiv
 Parameter estimate(s):
         Phi
0.0001428372
Fixed effects:  medida ~ poly(tiempo, 2)
                    Value Std.Error DF
(Intercept)       42.99444  2.185832 22
poly(tiempo, 2)1  36.94647  4.443447 22
poly(tiempo, 2)2 -28.30784  2.076632 22
                   t-value p-value
(Intercept)       19.669598       0
poly(tiempo, 2)1   8.314823       0
poly(tiempo, 2)2 -13.631610       0
 Correlation:
                 (Intr) p(,2)1
poly(tiempo, 2)1  0.828
poly(tiempo, 2)2 -0.475 -0.497
```

```
Standardized Within-Group Residuals:
       Min         Q1        Med         Q3
-1.0450552 -0.4346930 -0.1931843  0.5001057
       Max
 1.1924184

Number of Observations: 36
Number of Groups: 12
```

- Valor esperado de la constante y coeficientes, $\beta_0, \ldots, \beta_K$. También se conoce como los coeficientes fijos. Para obtener la tabla de sus estimaciones y los p-valores:

```
coef(summary(modelo))
```

```
                    Value Std.Error DF
(Intercept)       42.99444  2.185832 22
poly(tiempo, 2)1  36.94647  4.443447 22
poly(tiempo, 2)2 -28.30784  2.076632 22
                    t-value      p-value
(Intercept)       19.669598 1.886788e-15
poly(tiempo, 2)1   8.314823 3.092000e-08
poly(tiempo, 2)2 -13.631610 3.311649e-12
```

- Estimación de los efectos aleatorios, $(\beta_{0i}^*, \beta_{1i}^*, \beta_{2i}^*)$

```
ranef(modelo)
```

```
    (Intercept) poly(tiempo, 2)1
1  14.730441065       30.8722315
2   4.054046210       11.0739532
3  -5.907126225       -5.2717958
4   0.001779692        0.6665792
5   4.796479128       13.7400443
6  -7.601895827      -11.9016785
7  -8.829202865      -17.6630563
8   6.499328908        7.4984525
9   0.924045128       -6.5624802
10 -5.693831230      -15.1669218
11 -8.860625561       -8.7246544
12  5.886561577        1.4393264
   poly(tiempo, 2)2
1        -6.5833764
2        -1.2028619
3         2.8619041
4        -3.8718583
5        -6.6013406
6         1.3545235
```

```
7          7.7680601
8         -0.9466624
9          1.6842059
10         3.7578624
11         2.6200805
12        -0.8405368
```

Hay una fila para cada individuo.

La función `ranef` retorna $\hat{\theta}_{ki}$, donde $\beta_{ki} = \beta_k + \theta_{ki}$. Así pues, $\theta_{ki} \sim N\left(0, \sigma^2_{\beta_k}\right)$ se pueden interpretar como los **"efectos aleatorios centrados"** tal y como se ha escrito la ecuación del modelo.

- Matriz de covarianzas de la constante y coeficientes aleatorios, Ω:

```
getVarCov(modelo)
```

```
Random effects variance covariance matrix
                 (Intercept) poly(tiempo, 2)1
(Intercept)           56.642           96.553
poly(tiempo, 2)1      96.553          211.990
poly(tiempo, 2)2     -25.856          -54.998
                 poly(tiempo, 2)2
(Intercept)               -25.856
poly(tiempo, 2)1          -54.998
poly(tiempo, 2)2           26.816
  Standard Deviations: 7.5261 14.56 5.1784
```

- Matriz de correlaciones de los residuos, Σ_i

```
modelo$modelStruct$corStruct
```

```
Correlation structure of class corAR1 representing
         Phi
0.0001428372
```

Podemos especificar que la correlación entre efectos aleatorios sea cero con la función `pdDiag` en el argumento `random`:

```
modelo2 <- lme(fixed = medida ~ poly(tiempo, 2),
               data=datos,
               random = list(indiv=pdDiag(~ poly(tiempo, 2))),
               correlation = corAR1()
               )
summary(modelo2)
```

```
Linear mixed-effects model fit by REML
  Data: datos
       AIC      BIC    logLik
  217.8098 229.7819 -100.9049
```

```
Random effects:
 Formula: ~poly(tiempo, 2) | indiv
 Structure: Diagonal
        (Intercept) poly(tiempo, 2)1
StdDev:  0.00304907         9.921619
        poly(tiempo, 2)2 Residual
StdDev:     6.785107e-05 7.683502

Correlation Structure: AR(1)
 Formula: ~1 | indiv
 Parameter estimate(s):
     Phi
0.896943
Fixed effects:  medida ~ poly(tiempo, 2)
                    Value Std.Error DF
(Intercept)       42.99444  2.116748 22
poly(tiempo, 2)1  36.94647  4.443466 22
poly(tiempo, 2)2 -28.30784  2.065204 22
                   t-value p-value
(Intercept)       20.311555       0
poly(tiempo, 2)1   8.314786       0
poly(tiempo, 2)2 -13.707045       0
 Correlation:
                 (Intr) p(,2)1
poly(tiempo, 2)1  0.000
poly(tiempo, 2)2 -0.098  0.000

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3
-1.58347912 -0.87272384  0.04840927  0.77580090
        Max
 2.40352231

Number of Observations: 36
Number of Groups: 12
```

```
getVarCov(modelo2)
```

```
Random effects variance covariance matrix
                (Intercept) poly(tiempo, 2)1
(Intercept)       9.2968e-06            0.000
poly(tiempo, 2)1  0.0000e+00           98.439
poly(tiempo, 2)2  0.0000e+00            0.000
                poly(tiempo, 2)2
(Intercept)           0.0000e+00
```

```
poly(tiempo, 2)1       0.0000e+00
poly(tiempo, 2)2       4.6038e-09
  Standard Deviations: 0.0030491 9.9216 6.7851e-05
```

Y para contrastar esta asunción

```
anova(modelo, modelo2)
```

```
        Model df      AIC      BIC     logLik
modelo      1 11 208.6888 225.1504  -93.34439
modelo2     2  8 217.8098 229.7819 -100.90491
          Test  L.Ratio p-value
modelo
modelo2 1 vs 2 15.12105  0.0017
```

El mejor a escoger es el que contempla que hay correlación entre los efectos aleatorios.

Simplificación del modelo

Miramos primero si los coeficientes son aleatorios o fijos. Para ello comparamos el modelo completo con el modelo sólo con la constante aleatoria.

```
anova(modelo, update(modelo, random = ~ 1 | indiv))
```

```
                                    Model df
modelo                                  1 11
update(modelo, random = ~1 | indiv)     2  6
                                         AIC
modelo                                208.6888
update(modelo, random = ~1 | indiv) 214.6890
                                         BIC
modelo                                225.1504
update(modelo, random = ~1 | indiv) 223.6680
                                        logLik
modelo                                 -93.34439
update(modelo, random = ~1 | indiv) -101.34449
                                          Test
modelo
update(modelo, random = ~1 | indiv) 1 vs 2
                                       L.Ratio
modelo
update(modelo, random = ~1 | indiv) 16.00021
                                      p-value
modelo
update(modelo, random = ~1 | indiv)  0.0068
```

Con la función `anova` se comparan los dos modelos mediante el LRT, uno con los coeficientes aleatorios y el otro sólo con la constante aleatoria. En este caso, y como se ha dicho, el LRT para constrastar si las varianzas son cero no es del

todo adecuado. Existen otros tests basados en remuestreo, pero hasta la fecha no funcionan con `lme` y no se explicarán en este curso.

Basándonos en el LRT, y también el criterio AIC o BIC, se tiene que el modelo más complejo (el que supone que los coeficientes son aleatorios) es el que se eligirá.

Posteriormente miramos si la correlación entre los efectos aleatorios es cero o no:

```
anova(modelo, update(modelo, random=list(indiv=pdDiag(~poly(tiempo,2)))))
```

```
                                                                  Model
modelo                                                                1
update(modelo, random = list(indiv = pdDiag(~poly(tiempo, 2))))     2
                                                                     df
modelo                                                               11
update(modelo, random = list(indiv = pdDiag(~poly(tiempo, 2))))       8
                                                                    AIC
modelo                                                         208.6888
update(modelo, random = list(indiv = pdDiag(~poly(tiempo, 2)))) 217.8098
                                                                    BIC
modelo                                                         225.1504
update(modelo, random = list(indiv = pdDiag(~poly(tiempo, 2)))) 229.7819
                                                                 logLik
modelo                                                         -93.34439
update(modelo, random = list(indiv = pdDiag(~poly(tiempo, 2)))) -100.90491
                                                                   Test
modelo
update(modelo, random = list(indiv = pdDiag(~poly(tiempo, 2)))) 1 vs 2
                                                                L.Ratio
modelo
update(modelo, random = list(indiv = pdDiag(~poly(tiempo, 2)))) 15.12105
                                                                p-value
modelo
update(modelo, random = list(indiv = pdDiag(~poly(tiempo, 2))))  0.0017
```

Nos quedamos con el modelo más complejo, ya que el p-valor del LRT es $<$ 0.005.

Finalmente, miramos si podemos simplificar la matriz de correlación de los residuos. Comparamos mediante el LRT el modelo ajustado con uno que suponga independencia de los residuos:

```
anova(modelo, update(modelo, correlation=NULL))
```

```
                                  Model df
modelo                                1 11
update(modelo, correlation = NULL)    2 10
```

```
                                     AIC
modelo                             208.6888
update(modelo, correlation = NULL) 206.6888
                                     BIC
modelo                             225.1504
update(modelo, correlation = NULL) 221.6539
                                   logLik
modelo                             -93.34439
update(modelo, correlation = NULL) -93.34439
                                    Test
modelo
update(modelo, correlation = NULL) 1 vs 2
                                     L.Ratio
modelo
update(modelo, correlation = NULL) 3.668106e-10
                                   p-value
modelo
update(modelo, correlation = NULL)       1
```

Como el p-valor > 0.05, elegimos el modelo más simple (el de independencia de los residuos). Además, según el criteria AIC, o BIC (cuánto más bajo mejor), también nos decantamos por el modelo de independencia de los residuos.

```
modelo <- update(modelo, correlation=NULL)
```

En el siguiente paso evaluamos la significación de los efectos fijos:

```
coef(summary(modelo))
```

```
                    Value Std.Error DF
(Intercept)       42.99444  2.185834 22
poly(tiempo, 2)1  36.94647  4.443441 22
poly(tiempo, 2)2 -28.30784  2.076630 22
                   t-value      p-value
(Intercept)       19.669582 1.886821e-15
poly(tiempo, 2)1   8.314834 3.091930e-08
poly(tiempo, 2)2 -13.631623 3.311588e-12
```

Todos los coeficientes son significativos. Por lo tanto no podemos simplificar el modelo.

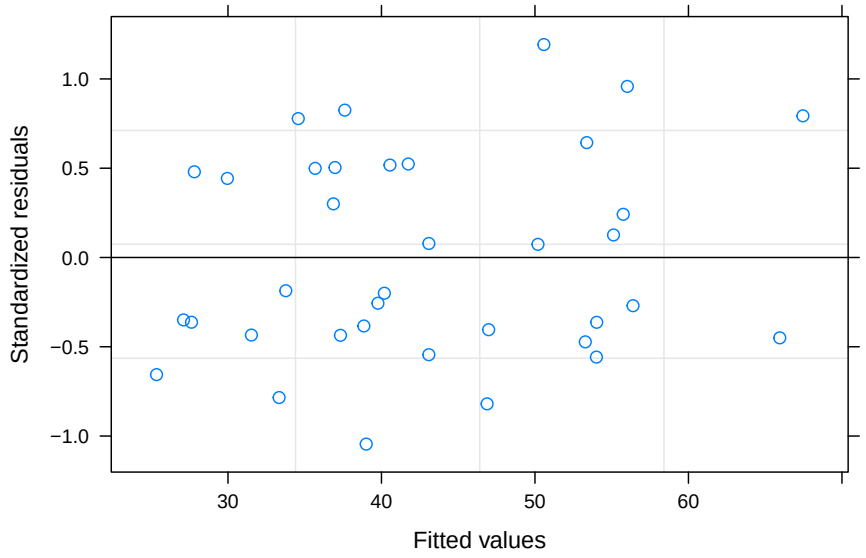
```
summary(modelo)
```

```
Linear mixed-effects model fit by REML
  Data: datos
       AIC      BIC    logLik
  206.6888 221.6539 -93.34439

Random effects:
```

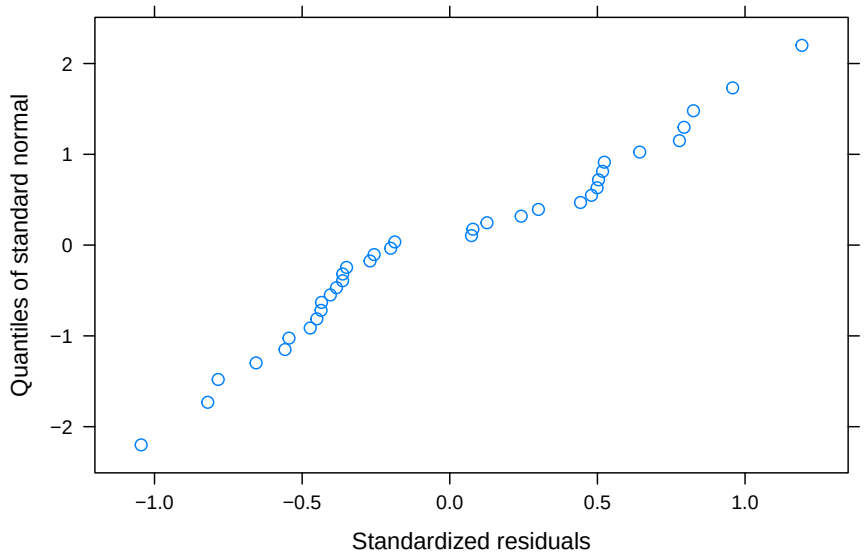
```
 Formula: ~poly(tiempo, 2) | indiv
 Structure: General positive-definite, Log-Cholesky parametrization
                 StdDev    Corr
(Intercept)       7.526075 (Intr) p(,2)1
poly(tiempo, 2)1 14.560044  0.881
poly(tiempo, 2)2  5.178188 -0.663 -0.729
Residual          1.441501

Fixed effects:  medida ~ poly(tiempo, 2)
                    Value Std.Error DF
(Intercept)       42.99444  2.185834 22
poly(tiempo, 2)1  36.94647  4.443441 22
poly(tiempo, 2)2 -28.30784  2.076630 22
                   t-value p-value
(Intercept)       19.669582       0
poly(tiempo, 2)1   8.314834       0
poly(tiempo, 2)2 -13.631623       0
 Correlation:
                 (Intr) p(,2)1
poly(tiempo, 2)1  0.828
poly(tiempo, 2)2 -0.475 -0.497

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3
-1.0451512 -0.4348223 -0.1931091  0.5002053
       Max
 1.1925501

Number of Observations: 36
Number of Groups: 12
```

Finalmente, validamos el modelo:

```
par(mfrow=c(1,2))
plot(modelo)
```




```
qqnorm(modelo)
```



Según estos gráficos, diremos que sí se cumplen las premisas sobre los residuos.

Predicciones:

Para calcular las predicciones nos será útil usar las funciones del paquete **ggeffects**. Con este paquete se pueden realizar las predicciones de distinto tipo y también graficarlas con el paquete ggplot2.

```
library(ggeffects)
```

```
pr.fixed <- ggpredict(modelo, "tiempo [all]", type="fixed")
pr.fixed
```

```
# Predicted values of medida

tiempo | Predicted |         95% CI
-----------------------------------
     1 |     32.12 | [29.28, 34.95]
     2 |     49.67 | [44.85, 54.48]
     3 |     47.20 | [41.58, 52.82]

Adjusted for:
* indiv = 0 (population-level)
```

```
pr.random <- ggpredict(modelo, "tiempo [all]", type="random")
pr.random
```

```
# Predicted values of medida

tiempo
------
     1
     2
     3

Adjusted for:
* indiv = 0 (population-level)
```

Con el argumento `type="random"`, el intervalo es más ancho porque no sólo tiene en cuenta el error estándar de las estimaciones de los parámetros sino también la varianza de los efectos aleatorios.

```
library(gridExtra)
grid.arrange(
    plot(pr.fixed) + ylim(25,60) + ggtitle("CI: fixed"),
    plot(pr.random) + ylim(25,60) + ggtitle("CI: random"),
nrow=1, ncol=2)
```

Resultado

Por lo tanto el modelo final contendrá el tiempo, el tiempo al cuadrado, la contaste y los coeficientes aleatorios. Finalmente, los residuos se puede suponer independientes.

Observaciones

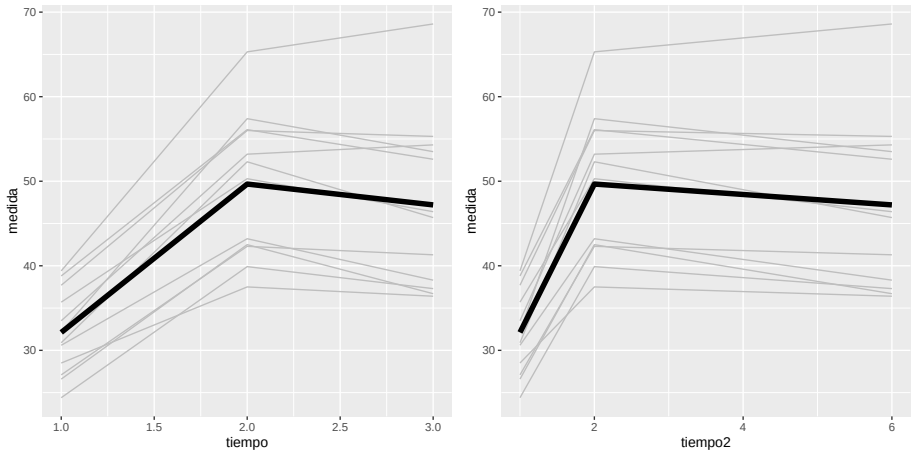
- Se pueden incorporar **términos splines** en la fórmula usando la función `ns` del paquete `spline`. Es útil cuando se tienen muchas medidas repetidas y/o en distintos momentos para los diferentes individuos. Se usa en las fórmulas (argumentos `fixed` i `random`)

```
lme(respuesta ~ ns(tiempo), random = ns(tiempo) | indiv, ...)
```

Comparación con las otras técnicas

A fin de poder comparar los resultados de los modelo LMM con los modelos basados en la suma de cuadrados y en la respuesta multivariante, el tiempo se debe tratar como factor. Fíjate en el uso de `as.factor` para convertir una variable numérica a factor o variable categórica. Para ello, hay que tener las mismas categorías de tiempo para todos los individuos. Además, tanto los modelos de respuesta multivariable como los basados en suma de cuadrados, asumen la pendiente fija o constante, y la correlación sin estructura.

Nota: Es importante notar que las técnicas de respuesta multivariable y de suma de cuadrados al tratar la variable tiempo como factor, no se puede distinguir si tiempos de las medidas son o no equiespaiados. Por ejemplo los resultados obtenidos mediante estas dos técnicas serán los mismos tanto si se recogen las medidas a 1h, 2h y 3h, o si se recoge a 1h, 2h y 6h. En cambio, si se desea estudiar el efecto lineal ambas situaciones son muy distintas.



```
modelo <- lme(fixed = medida ~ as.factor(tiempo),
              data=datos,
              random = ~ 1 | indiv,
```

```
              correlation = corSymm()
              )
summary(modelo)
```

```
Linear mixed-effects model fit by REML
  Data: datos
       AIC      BIC    logLik
  218.5233 230.4954 -101.2616

Random effects:
 Formula: ~1 | indiv
        (Intercept) Residual
StdDev:    7.069756 3.498312

Correlation Structure: General
 Formula: ~1 | indiv
 Parameter estimate(s):
 Correlation:
  1      2
2  0.081
3 -0.561  0.628
Fixed effects:  medida ~ as.factor(tiempo)
                     Value Std.Error DF
(Intercept)        32.11667  2.277053 22
as.factor(tiempo)2 17.55000  1.368874 22
as.factor(tiempo)3 15.08333  1.784415 22
                     t-value p-value
(Intercept)        14.104488       0
as.factor(tiempo)2 12.820757       0
as.factor(tiempo)3  8.452817       0
 Correlation:
                   (Intr) as.()2
as.factor(tiempo)2 -0.301
as.factor(tiempo)3 -0.392  0.880

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3
-1.95466422 -0.48588298 -0.07605957  0.47680669
        Max
 2.08061488

Number of Observations: 36
Number of Groups: 12
```

```
coef(summary(modelo))
```

```
                     Value Std.Error DF
(Intercept)        32.11667  2.277053 22
as.factor(tiempo)2 17.55000  1.368874 22
as.factor(tiempo)3 15.08333  1.784415 22
                    t-value      p-value
(Intercept)        14.104488 1.683824e-12
as.factor(tiempo)2 12.820757 1.104451e-11
as.factor(tiempo)3  8.452817 2.338498e-08
```

La función `anova` aplicada a un sólo modelo ajustado es útil para contrastar la significación de un factor de más de una categoría (o posibles interacciones de factores de más de dos categorías).

```
anova(modelo)
```

```
                  numDF denDF  F-value p-value
(Intercept)           1    22 319.4667  <.0001
as.factor(tiempo)     2    22  99.9590  <.0001
```

6.7.2 Ejemplo 2

Analicemos ahora los datos introducidos en el capítulo anterior en el ejercicio 2. Recordemos que en la base de datos "Ejemplo_1W1B.csv" se tienen los datos de un estudio en el que participan 24 individuos randomizados en dos grupos de tratamiento (`trat`). Como en el anterior ejemplo, para cada invidivuo se miden los niveles a 1, 2 y 3 horas.

```
datos <- read.csv2("datos/Ejemplo_1W1B.csv")
```

Como antes, ordenamos por individuo (de 1 a 24) y por tiempo

```
datos <- arrange(datos, indiv2, tiempo)
```

```
modelo <- lme(fixed = medida ~ poly(tiempo,2)*trat,
              data=datos,
              random = ~ poly(tiempo,2) | indiv2,
              correlation = corAR1()
              )
summary(modelo)
```

```
Linear mixed-effects model fit by REML
  Data: datos
      AIC      BIC   logLik
  386.914 417.5692 -179.457

Random effects:
```

```
 Formula: ~poly(tiempo, 2) | indiv2
 Structure: General positive-definite, Log-Cholesky parametrization
                StdDev    Corr
(Intercept)      6.528818 (Intr) p(,2)1
poly(tiempo, 2)1 15.016144  0.720
poly(tiempo, 2)2  5.983554 -0.637 -0.653
Residual          1.290085

Correlation Structure: AR(1)
 Formula: ~1 | indiv2
 Parameter estimate(s):
        Phi
5.458068e-05
Fixed effects:  medida ~ poly(tiempo, 2) * trat
                         Value Std.Error DF
(Intercept)           22.16667  4.241672 44
poly(tiempo, 2)1     -35.16063 10.516421 44
poly(tiempo, 2)2      33.10000  5.617817 44
trat                  10.41389  2.682669 22
poly(tiempo, 2)1:trat 43.70542  6.651169 44
poly(tiempo, 2)2:trat -36.56667  3.553020 44
                        t-value p-value
(Intercept)            5.225927  0.0000
poly(tiempo, 2)1      -3.343403  0.0017
poly(tiempo, 2)2       5.891968  0.0000
trat                   3.881914  0.0008
poly(tiempo, 2)1:trat  6.571088  0.0000
poly(tiempo, 2)2:trat -10.291715  0.0000
 Correlation:
                      (Intr) pl(,2)1 pl(,2)2
poly(tiempo, 2)1       0.659
poly(tiempo, 2)2      -0.435 -0.414
trat                  -0.949 -0.625   0.413
poly(tiempo, 2)1:trat -0.625 -0.949   0.393
poly(tiempo, 2)2:trat  0.413  0.393  -0.949
                      trat   p(,2)1:
poly(tiempo, 2)1
poly(tiempo, 2)2
trat
poly(tiempo, 2)1:trat  0.659
poly(tiempo, 2)2:trat -0.435 -0.414

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3
-1.17636054 -0.41451565 -0.05512438  0.37021883
        Max
```

```
 1.36156358

Number of Observations: 72
Number of Groups: 24
```

Nota: Si los individuos estuvieran anidados dentro de clusters, se especificaría en el argumento `random = ~ 1 | indiv / clusters`, donde "cluster" sería el nombre de la variable que codifica los clusters.

Observación Para que el modelo quede bien definido no es posible poner la interacción del tiempo y el tratamiento como coeficiente aleatorio. De esta manera se especifican como aleatorios la costante y los coeficientes del tiempo (lineal y cuadrático) para el grupo control.

Como en el anterior ejemplo, contrastamos la significación de los coeficientes aleatorios del tiempo:

```
anova(modelo, update(modelo, random = ~ 1 | indiv2))
```

```
                                    Model df
modelo                                  1 14
update(modelo, random = ~1 | indiv2)    2  9
                                         AIC
modelo                                386.9140
update(modelo, random = ~1 | indiv2) 393.6099
                                         BIC
modelo                                417.5692
update(modelo, random = ~1 | indiv2) 413.3168
                                       logLik
modelo                                -179.457
update(modelo, random = ~1 | indiv2) -187.805
                                       Test
modelo
update(modelo, random = ~1 | indiv2) 1 vs 2
                                     L.Ratio
modelo
update(modelo, random = ~1 | indiv2) 16.69594
                                     p-value
modelo
update(modelo, random = ~1 | indiv2)  0.0051
```

Según el criterio AIC o BIC, el modelo con pendientes aleatorias es mejor.

Luego, contrastamos si se puede simplificar la matriz de correlaciones de los efectos aleatorios:

```
anova(modelo, update(modelo, random = list(indiv2=pdDiag(~poly(tiempo,2)))))
```

```
                                                                    Model
```

```
modelo                                                                  1
update(modelo, random = list(indiv2 = pdDiag(~poly(tiempo, 2))))        2
                                                                       df
modelo                                                                 14
update(modelo, random = list(indiv2 = pdDiag(~poly(tiempo, 2))))       11
                                                                      AIC
modelo                                                           386.9140
update(modelo, random = list(indiv2 = pdDiag(~poly(tiempo, 2))))  397.1845
                                                                      BIC
modelo                                                           417.5692
update(modelo, random = list(indiv2 = pdDiag(~poly(tiempo, 2))))  421.2707
                                                                   logLik
modelo                                                           -179.4570
update(modelo, random = list(indiv2 = pdDiag(~poly(tiempo, 2))))  -187.5922
                                                                     Test
modelo
update(modelo, random = list(indiv2 = pdDiag(~poly(tiempo, 2))))  1 vs 2
                                                                  L.Ratio
modelo
update(modelo, random = list(indiv2 = pdDiag(~poly(tiempo, 2))))  16.27047
                                                                  p-value
modelo
update(modelo, random = list(indiv2 = pdDiag(~poly(tiempo, 2))))    0.001
```

El test LRT es significativo (p-valor < 0.05). Por lo tanto nos quedamos con el modelo más complejo que supone que hay correlación entre los efectos aleatorios.

Seguidamente, miramos si se puede simplificar la matriz de correlaciones de los errores.

```
anova(modelo, update(modelo, correlation=NULL))
```

```
                                    Model df
modelo                                 1 14
update(modelo, correlation = NULL)     2 13
                                       AIC
modelo                              386.914
update(modelo, correlation = NULL) 384.914
                                        BIC
modelo                              417.5692
update(modelo, correlation = NULL) 413.3795
                                     logLik
modelo                              -179.457
update(modelo, correlation = NULL) -179.457
                                      Test
modelo
update(modelo, correlation = NULL) 1 vs 2
```

```
                                      L.Ratio
modelo
update(modelo, correlation = NULL) 1.063989e-08
                                     p-value
modelo
update(modelo, correlation = NULL)  0.9999
```

Sí que se puede suponer que hay indipendencia entre los residuos.

```
modelo <- update(modelo, correlation=NULL)
```

Por lo tanto el modelo final, que supone independencia entre residuos, tiene la siguiente estimación de los efectos fijos:

```
coef(summary(modelo))
```

```
                         Value Std.Error DF
(Intercept)           22.16667  4.241667 44
poly(tiempo, 2)1     -35.16063 10.516526 44
poly(tiempo, 2)2      33.10000  5.617838 44
trat                  10.41389  2.682666 22
poly(tiempo, 2)1:trat 43.70542  6.651235 44
poly(tiempo, 2)2:trat -36.56667  3.553033 44
                        t-value      p-value
(Intercept)            5.225932 4.556683e-06
poly(tiempo, 2)1      -3.343370 1.698154e-03
poly(tiempo, 2)2       5.891946 4.863408e-07
trat                   3.881918 8.040480e-04
poly(tiempo, 2)1:trat  6.571023 4.876727e-08
poly(tiempo, 2)2:trat -10.291677 2.732020e-13
```

Vemos como el efecto del tiempo para el grupo control no llega a ser significativo (p-valores >0.05) tanto para su componente lineal como cuadrático. Hay efecto del tratamiento en el momento basal (`trat2`).

Si queremos ver el efecto del tiempo para el grupo 2, hay que cambiar su categoría de referencia.

```
datos$trat <- relevel(factor(datos$trat),2)
coef(summary(update(modelo)))
```

```
                          Value Std.Error DF
(Intercept)            42.99444  1.896932 44
poly(tiempo, 2)1       52.25020  4.703133 44
poly(tiempo, 2)2      -40.03333  2.512374 44
trat1                 -10.41389  2.682666 22
poly(tiempo, 2)1:trat1 -43.70542  6.651235 44
poly(tiempo, 2)2:trat1  36.56667  3.553033 44
                         t-value      p-value
```

```
(Intercept)             22.665259 6.756030e-26
poly(tiempo, 2)1        11.109658 2.358252e-14
poly(tiempo, 2)2       -15.934465 6.870764e-20
trat1                   -3.881917 8.040486e-04
poly(tiempo, 2)1:trat1  -6.571023 4.876723e-08
poly(tiempo, 2)2:trat1  10.291676 2.732023e-13
```

Vemos que para el grupo 2 tanto la componente lineal como la cuadrática del tiempo son significativas.

Con la siguiente matriz de varianzas y covarianzas de los efectos aleatorios:

```
getVarCov(modelo)
```

```
Random effects variance covariance matrix
                (Intercept) poly(tiempo, 2)1
(Intercept)          42.625           70.566
poly(tiempo, 2)1     70.566          225.490
poly(tiempo, 2)2    -24.889          -58.669
                poly(tiempo, 2)2
(Intercept)              -24.889
poly(tiempo, 2)1         -58.669
poly(tiempo, 2)2          35.802
  Standard Deviations: 6.5288 15.016 5.9835
```

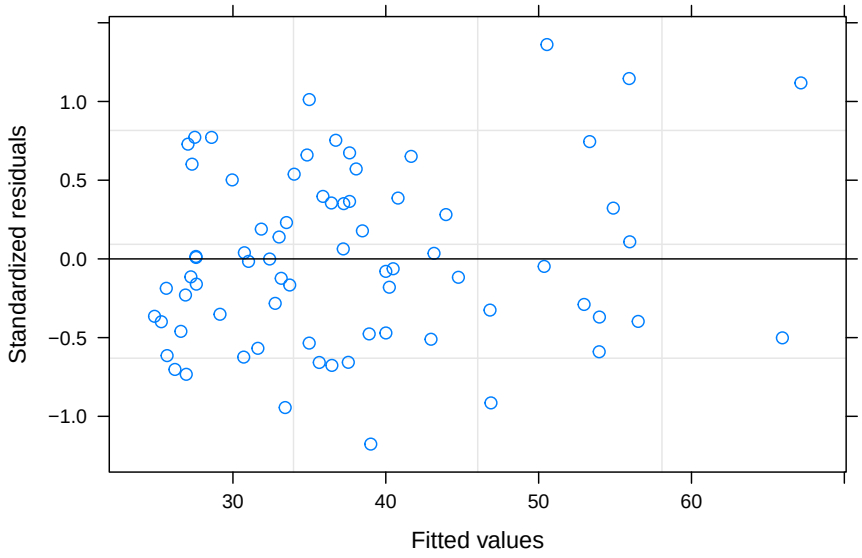
Y varianza de los residuos

```
sigma(modelo)^2
```

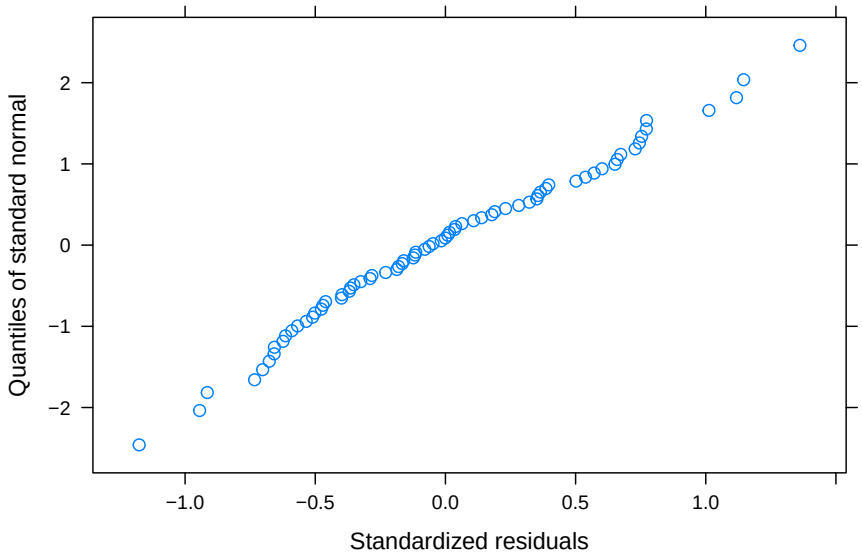
```
[1] 1.664273
```

Por último, validamos el modelo

```
par(mfrow=c(1,2))
plot(modelo)
```




```
qqnorm(modelo)
```



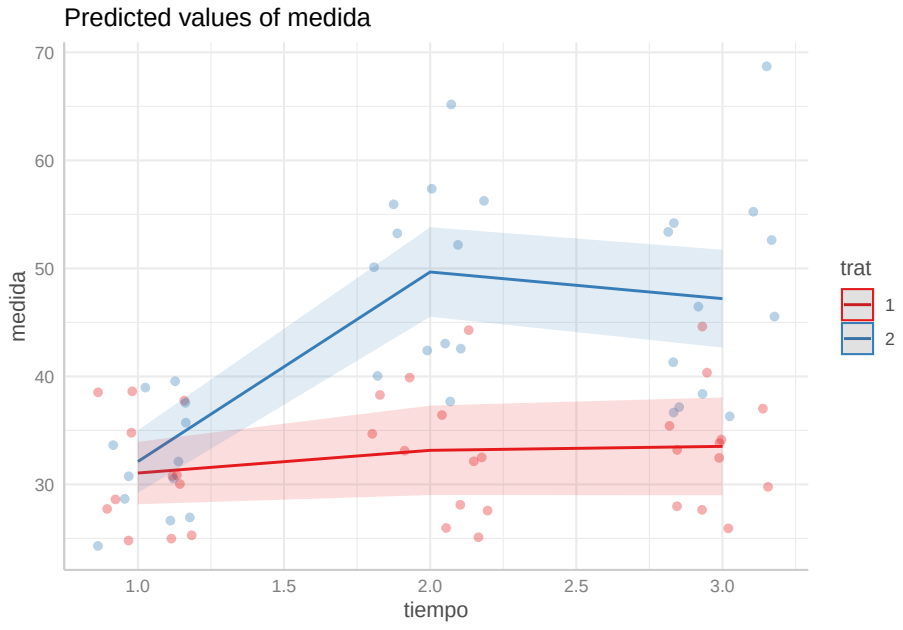
Según los gráficos, parece que sí que se cumplen las premisas sobre los residuos.

Predicciones

```
pr <- ggpredict(modelo, terms = c("tiempo [all]", "trat"))
pr
```

```
# Predicted values of medida

# trat = 1

tiempo | Predicted |         95% CI
-----------------------------------
     1 |     31.06 | [28.17, 33.95]
     2 |     33.16 | [29.02, 37.30]
     3 |     33.52 | [29.00, 38.05]

# trat = 2

tiempo | Predicted |         95% CI
-----------------------------------
     1 |     32.12 | [29.22, 35.01]
     2 |     49.67 | [45.53, 53.81]
     3 |     47.20 | [42.67, 51.73]

Adjusted for:
* indiv2 = 0 (population-level)
```

```
plot(pr, add.data = TRUE)
```



6.7.3 Ejemplo 3

Finalmente, analicemos ahora los datos introducidos en el capítulo anterior en el ejercicio 3. En un estudio se quieren comparar el efecto de régimen de ejercicio sobre el sobrepeso. Para ello se reclutan 100 personas. A la mitad se le asigna el régimen y al resto se le hacen algunas recomendaciones (grupo control). Se mide el índice de masa corporal justo antes de empezar el estudio (momento basal), y al cabo de 1, 2 y 3 semanas. Como la edad es una variable importante para predecir el IMC también se registra.

Los datos los encontrarás en el fichero "imc.csv"

En este ejemplo, vemos como en algunos de los individuos nos falta alguna medida en a partir de la primera semana. Por este motivo usaremos la técnica de los LMM.

```
datos <- read.csv2("datos/imc.csv")
```

Nos aseguramos que los datos estén ordenados por individuo y tiempo

```
datos <- arrange(datos, indiv, tiempo)
```

Recodificamos nuestra variable tratamiento:

```
datos$tx <- factor(datos$tx, 1:2, c("Control", "Tratados"))
summary(datos)
```

```
   respuesta         indiv            tiempo
 Min.   : 9.80   Min.   :  1.00   Min.   :0.00
 1st Qu.:27.02   1st Qu.: 25.75   1st Qu.:0.75
 Median :30.75   Median : 50.50   Median :1.50
 Mean   :30.46   Mean   : 50.50   Mean   :1.50
 3rd Qu.:34.60   3rd Qu.: 75.25   3rd Qu.:2.25
 Max.   :43.70   Max.   :100.00   Max.   :3.00
 NA's   :50
      edad              tx
 Min.   :25.00   Control :200
 1st Qu.:43.00   Tratados:200
 Median :49.00
 Mean   :49.03
 3rd Qu.:57.00
 Max.   :69.00
```

Recordemos también que estos datos no puedieron ser analizados con métodos tradicionales ya que tienen inviduos con datos faltantes

```
sum(with(datos, tapply(is.na(respuesta), indiv, any)))
```

```
[1] 42
```

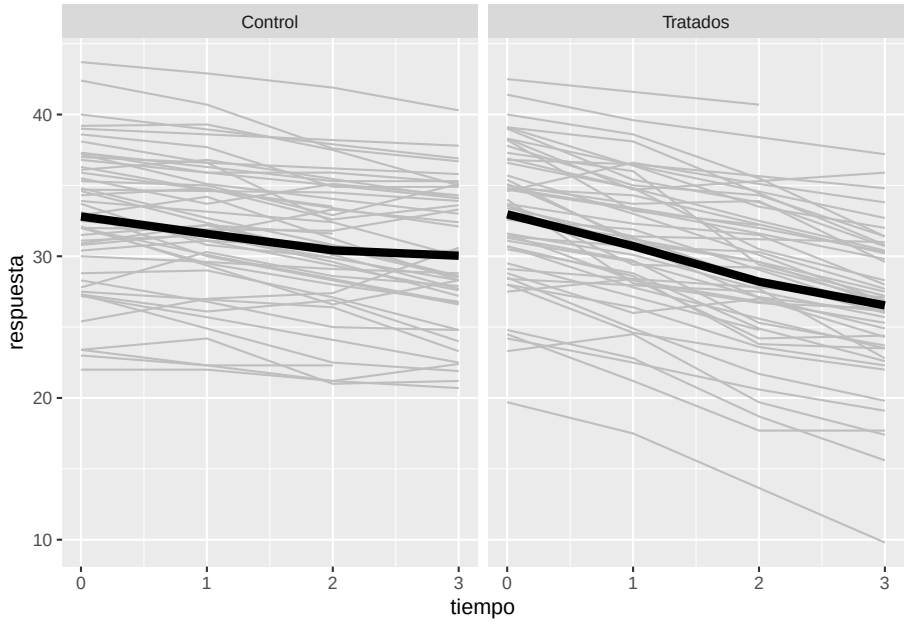
Elimino las observaciones con valores missing (que no los individuos!)

```
datos <- subset(datos, !is.na(respuesta))
# número de individuos con 2, 3 o 4 medidas válidas.
table(table(datos$indiv))
```

```
 2  3  4
 8 34 58
```

Como siempre, hagamos una visualización de los datos

```
library(ggplot2)
p <- ggplot(data = datos, aes(x = tiempo, y = respuesta, group = indiv))
p <- p + geom_line(col="grey") + stat_summary(aes(group = 1),
    geom = "line", fun = mean, size=2)
p + facet_grid( ~ tx)
```



6.7.3.1 Análisis del grupo control

Si analizamos sólo el grupo control, se trata de un diseño 1W con una covariable (edad).

```
datos <- subset(datos, tx=='Control')
datos <- na.omit(datos)
```

Ajustamos el modelo más completo, con la edad y el tiempo hasta el término cúbico ya que tenemos cuatro medidas.

```
library(nlme)
modelo <- lme(respuesta ~ poly(tiempo,3) + edad,
              random= ~ poly(tiempo,3) | indiv,
              data=datos,
              #correlation = corSymm(), # sin estructura
              correlation=corCAR1(form = ~ tiempo | indiv),
              control=lmeControl(opt="optim"))
summary(modelo)
```

```
Linear mixed-effects model fit by REML
  Data: datos
       AIC      BIC    logLik
  672.3888 724.9826 -319.1944

Random effects:
 Formula: ~poly(tiempo, 3) | indiv
 Structure: General positive-definite, Log-Cholesky parametrization
                 StdDev     Corr
(Intercept)       2.1235727 (Intr) p(,3)1 p(,3)2
poly(tiempo, 3)1 10.8729435 -0.143
poly(tiempo, 3)2  5.3455110  0.317  0.071
poly(tiempo, 3)3  4.2631158  0.057 -0.026  0.193
Residual          0.5431964

Correlation Structure: Continuous AR(1)
 Formula: ~tiempo | indiv
 Parameter estimate(s):
      Phi
0.1999231
Fixed effects:  respuesta ~ poly(tiempo, 3) + edad
                     Value Std.Error  DF
(Intercept)       10.409728 1.4770932 115
poly(tiempo, 3)1 -15.045596 1.6657189 115
poly(tiempo, 3)2  -0.518758 1.0075975 115
poly(tiempo, 3)3   1.865008 0.8918255 115
edad               0.432945 0.0297828  48
                   t-value p-value
(Intercept)       7.047442  0.0000
poly(tiempo, 3)1 -9.032494  0.0000
poly(tiempo, 3)2 -0.514846  0.6076
poly(tiempo, 3)3  2.091225  0.0387
edad             14.536718  0.0000
 Correlation:
                 (Intr) p(,3)1 p(,3)2 p(,3)3
poly(tiempo, 3)1 -0.024
```

```
poly(tiempo, 3)2  0.081  0.089
poly(tiempo, 3)3  0.012 -0.041  0.149
edad             -0.978 -0.001 -0.037 -0.003

Standardized Within-Group Residuals:
         Min           Q1          Med
-1.025920132 -0.293663743 -0.008367651
          Q3          Max
 0.290693758  1.093099300

Number of Observations: 168
Number of Groups: 50
```

El modelo de correlación sin estructura no converge (NOTA: intentad ejecutar el mismo código cambiando la correlación que tiene por la que está comentada). Es normal ya que tenemos distintas medidas.

Es importante especificar la AR(1) continua `corCAR1` ya que tenemos algunos individuos con datos faltantes en algunas de sus medidas. Luego el tiempo que ha pasado entre las medidas disponibles hay que tenerlo en cuenta.

Fíjate en la varianza de los efectos aleatorios, sobretodo en la constante si en el modelo no ponemos la edad,

```
modelo0 <- update(modelo, fixed = . ~ . -edad)
getVarCov(modelo)
```

```
Random effects variance covariance matrix
                (Intercept) poly(tiempo, 3)1
(Intercept)          4.5096          -3.2917
poly(tiempo, 3)1    -3.2917         118.2200
poly(tiempo, 3)2     3.6035           4.1274
poly(tiempo, 3)3     0.5144          -1.1839
                poly(tiempo, 3)2
(Intercept)               3.6035
poly(tiempo, 3)1          4.1274
poly(tiempo, 3)2         28.5740
poly(tiempo, 3)3          4.4056
                poly(tiempo, 3)3
(Intercept)               0.5144
poly(tiempo, 3)1         -1.1839
poly(tiempo, 3)2          4.4056
poly(tiempo, 3)3         18.1740
  Standard Deviations: 2.1236 10.873 5.3455 4.2631
```

```
getVarCov(modelo0) # sin la edad
```

```
Random effects variance covariance matrix
```

```
                (Intercept) poly(tiempo, 3)1
(Intercept)         22.65700         -2.56910
poly(tiempo, 3)1    -2.56910        108.79000
poly(tiempo, 3)2     0.79669          2.88170
poly(tiempo, 3)3     0.60630          0.23627
                poly(tiempo, 3)2
(Intercept)              0.79669
poly(tiempo, 3)1         2.88170
poly(tiempo, 3)2        20.71700
poly(tiempo, 3)3         3.90920
                poly(tiempo, 3)3
(Intercept)              0.60630
poly(tiempo, 3)1         0.23627
poly(tiempo, 3)2         3.90920
poly(tiempo, 3)3        13.19500
  Standard Deviations: 4.76 10.43 4.5516 3.6326
```

Es importante poner la edad ya que si no, la varianza de los efectos aleatorios quedan infladas y la inferencia no es válida.

Contraste de los efectos aleatorios

```
anova(modelo, update(modelo, random= ~ 1 | indiv))
```

```
                                   Model df
modelo                                 1 17
update(modelo, random = ~1 | indiv)    2  8
                                        AIC
modelo                             672.3888
update(modelo, random = ~1 | indiv) 662.1079
                                        BIC
modelo                             724.9826
update(modelo, random = ~1 | indiv) 686.8579
                                     logLik
modelo                             -319.1944
update(modelo, random = ~1 | indiv) -323.0539
                                      Test
modelo
update(modelo, random = ~1 | indiv) 1 vs 2
                                    L.Ratio
modelo
update(modelo, random = ~1 | indiv) 7.719062
                                   p-value
modelo
update(modelo, random = ~1 | indiv)  0.5627
```

Nos quedamos con el modelo con los coeficientes del tiempo fijos:

```
modelo <- update(modelo, random= ~ 1 | indiv)
```

> **Observación** Fíjate en el valor de ϕ de la matriz de correlaciones de los errores: al considerar el coeficiente del tiempo como fijo, ha pasado de ser cero a un valor alto. Al considerar el coeficiente aleatorio en cierta manera se inducía una estructura de AR entre las observaciones y ya no hacía falta considerar los errores correlacionados. Por esto, el orden en que se simplifica el modelo es importante.

Si queremos **contrastar si la constante es aletoria** se compara el modelo con todos los efectos fijos. Así que ya no se podrá usar la función `lme` sino que se usará la función `gls` en su lugar.

```
modelo.gls <- gls(respuesta ~ poly(tiempo,3) + edad,
                  data=datos,
                  correlation=corCAR1(form = ~ tiempo |indiv))
summary(modelo.gls)
```

```
Generalized least squares fit by REML
  Model: respuesta ~ poly(tiempo, 3) + edad
  Data: datos
       AIC      BIC    logLik
  660.2088 681.8651 -323.1044

Correlation Structure: Continuous AR(1)
 Formula: ~tiempo | indiv
 Parameter estimate(s):
      Phi
0.8409838

Coefficients:
                     Value Std.Error   t-value
(Intercept)       10.743564 1.5857081  6.775247
poly(tiempo, 3)1 -15.049949 1.6527760 -9.105861
poly(tiempo, 3)2  -0.382757 1.0563295 -0.362347
poly(tiempo, 3)3   1.978662 0.8789044  2.251283
edad               0.425951 0.0319656 13.325311
                  p-value
(Intercept)        0.0000
poly(tiempo, 3)1   0.0000
poly(tiempo, 3)2   0.7176
poly(tiempo, 3)3   0.0257
edad               0.0000

 Correlation:
                  (Intr) p(,3)1 p(,3)2 p(,3)3
```

```
poly(tiempo, 3)1  0.001
poly(tiempo, 3)2 -0.022  0.019
poly(tiempo, 3)3  0.003 -0.121  0.024
edad             -0.980 -0.002 -0.008 -0.001

Standardized residuals:
        Min          Q1         Med          Q3
-2.39865570 -0.70572628 -0.01464924  0.74989249
        Max
 2.07647314

Residual standard error: 2.492824
Degrees of freedom: 168 total; 163 residual
```

```
summary(modelo)
```

```
Linear mixed-effects model fit by REML
  Data: datos
       AIC      BIC    logLik
  662.1079 686.8579 -323.0539

Random effects:
 Formula: ~1 | indiv
        (Intercept) Residual
StdDev:   0.8028047 2.378488

Correlation Structure: Continuous AR(1)
 Formula: ~tiempo | indiv
 Parameter estimate(s):
      Phi
0.8266025
Fixed effects:  respuesta ~ poly(tiempo, 3) + edad
                     Value Std.Error  DF
(Intercept)       10.738507 1.6033545 115
poly(tiempo, 3)1 -15.048689 1.6348726 115
poly(tiempo, 3)2  -0.382488 1.0548435 115
poly(tiempo, 3)3   1.978466 0.8786767 115
edad               0.426057 0.0323229  48
                   t-value p-value
(Intercept)       6.697525  0.0000
poly(tiempo, 3)1 -9.204808  0.0000
poly(tiempo, 3)2 -0.362602  0.7176
poly(tiempo, 3)3  2.251643  0.0262
edad             13.181289  0.0000
 Correlation:
                 (Intr) p(,3)1 p(,3)2 p(,3)3
```

```
poly(tiempo, 3)1  0.001
poly(tiempo, 3)2 -0.020  0.019
poly(tiempo, 3)3  0.003 -0.122  0.024
edad             -0.980 -0.002 -0.008 -0.001

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3
-2.25670886 -0.66026076  0.01991706  0.66013218
        Max
 1.92440305

Number of Observations: 168
Number of Groups: 50
```

```
anova(modelo, modelo.gls)
```

```
           Model df      AIC      BIC    logLik
modelo         1  8 662.1079 686.8579 -323.0539
modelo.gls     2  7 660.2088 681.8651 -323.1044
             Test   L.Ratio p-value
modelo
modelo.gls 1 vs 2 0.1009386  0.7507
```

Los dos modelos no están anidados. Así que a parte del p-valor del LRT también miraremos el AIC y el BIC. Bajo los tres criterios nos decantamos por el modelo con la constante aleatoria.

Estructura de correlación de los errores

Comparamos con la matriz de independencia

```
anova(modelo, update(modelo, correlation=NULL))
```

```
                                    Model df
modelo                                  1  8
update(modelo, correlation = NULL)      2  7
                                         AIC
modelo                                662.1079
update(modelo, correlation = NULL) 677.6130
                                         BIC
modelo                                686.8579
update(modelo, correlation = NULL) 699.2693
                                       logLik
modelo                                -323.0539
update(modelo, correlation = NULL) -331.8065
                                      Test
modelo
update(modelo, correlation = NULL) 1 vs 2
```

```
                                    L.Ratio
modelo
update(modelo, correlation = NULL) 17.50515
                                   p-value
modelo
update(modelo, correlation = NULL)  <.0001
```

Nos quedamos con la estructura AR1

Contraste de los efectos fijos

```
coef(summary(modelo))
```

```
                      Value Std.Error  DF
(Intercept)       10.7385066 1.6033545 115
poly(tiempo, 3)1 -15.0486889 1.6348726 115
poly(tiempo, 3)2  -0.3824884 1.0548435 115
poly(tiempo, 3)3   1.9784662 0.8786767 115
edad               0.4260575 0.0323229  48
                   t-value      p-value
(Intercept)       6.697525 8.187045e-10
poly(tiempo, 3)1 -9.204808 1.848306e-15
poly(tiempo, 3)2 -0.362602 7.175678e-01
poly(tiempo, 3)3  2.251643 2.624396e-02
edad             13.181289 1.441756e-17
```

Vemos como la parte cuadrática no es significativa y la cúbico tampoco. Para contrastar los dos términos (cuadrático y cúbico) a la vez comparamos mediante el LRT el modelo completo con el modelo que supone el efecto del tiempo lineal

```
anova(
  update(modelo, method="ML"),
  update(modelo, fixed = . ~ . - poly(tiempo,3) + tiempo, method="ML")
)
```

```
                                                                    Model
update(modelo, method = "ML")                                           1
update(modelo, fixed = . ~ . - poly(tiempo, 3) + tiempo, method = "ML")     2
                                                                       df
update(modelo, method = "ML")                                           8
update(modelo, fixed = . ~ . - poly(tiempo, 3) + tiempo, method = "ML")  6
                                                                      AIC
update(modelo, method = "ML")                                    662.6555
update(modelo, fixed = . ~ . - poly(tiempo, 3) + tiempo, method = "ML") 663.4848
                                                                      BIC
update(modelo, method = "ML")                                    687.6473
update(modelo, fixed = . ~ . - poly(tiempo, 3) + tiempo, method = "ML") 682.2286
                                                                   logLik
update(modelo, method = "ML")                                   -323.3278
```

```
update(modelo, fixed = . ~ . - poly(tiempo, 3) + tiempo, method = "ML") -325.7424
                                                                          Test
update(modelo, method = "ML")
update(modelo, fixed = . ~ . - poly(tiempo, 3) + tiempo, method = "ML") 1 vs 2
                                                                        L.Ratio
update(modelo, method = "ML")
update(modelo, fixed = . ~ . - poly(tiempo, 3) + tiempo, method = "ML") 4.82924
                                                                        p-value
update(modelo, method = "ML")
update(modelo, fixed = . ~ . - poly(tiempo, 3) + tiempo, method = "ML")  0.0894
```

El p-valor > 0.05, por lo tanto nos quedamos con el modelo lineal.

```
modelo <- update(modelo, fixed = . ~ . - poly(tiempo,3) + tiempo)
summary(modelo)
```

```
Linear mixed-effects model fit by REML
  Data: datos
       AIC      BIC    logLik
  671.7705 690.4062 -329.8853

Random effects:
 Formula: ~1 | indiv
        (Intercept) Residual
StdDev:     1.61273 1.877586

Correlation Structure: Continuous AR(1)
 Formula: ~tiempo | indiv
 Parameter estimate(s):
      Phi
0.7045807
Fixed effects:  respuesta ~ edad + tiempo
                Value Std.Error  DF   t-value
(Intercept) 12.084507 1.5894665 117  7.602870
edad         0.426773 0.0319158  48 13.371839
tiempo      -0.959328 0.1030230 117 -9.311792
            p-value
(Intercept)       0
edad              0
tiempo            0
 Correlation:
       (Intr) edad
edad   -0.976
tiempo -0.092 -0.002

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3
```

```
-2.13574976 -0.49274745  0.00586512  0.57010471
        Max
 2.16657427

Number of Observations: 168
Number of Groups: 50
```

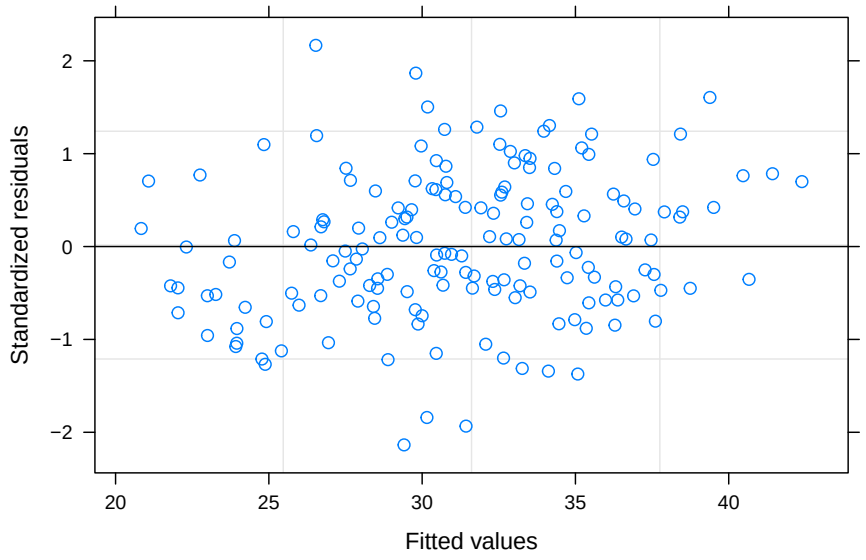
```
coef(summary(modelo))
```

```
                Value  Std.Error  DF   t-value
(Intercept) 12.0845065 1.58946650 117  7.602870
edad         0.4267727 0.03191578  48 13.371839
tiempo      -0.9593285 0.10302297 117 -9.311792
                p-value
(Intercept) 7.939399e-12
edad        8.365041e-18
tiempo      9.105728e-16
```

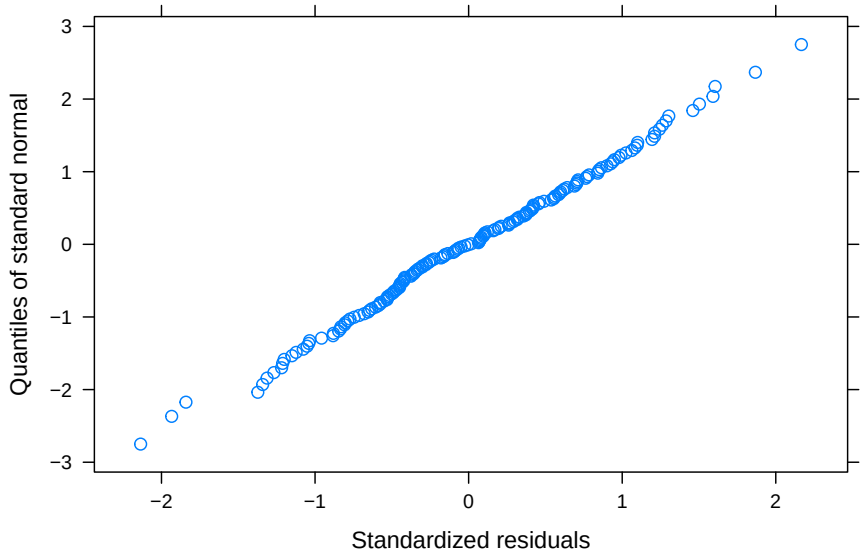
Validación del modelo

- Errores

```
par(mfrow=c(1,2))
plot(modelo)
```

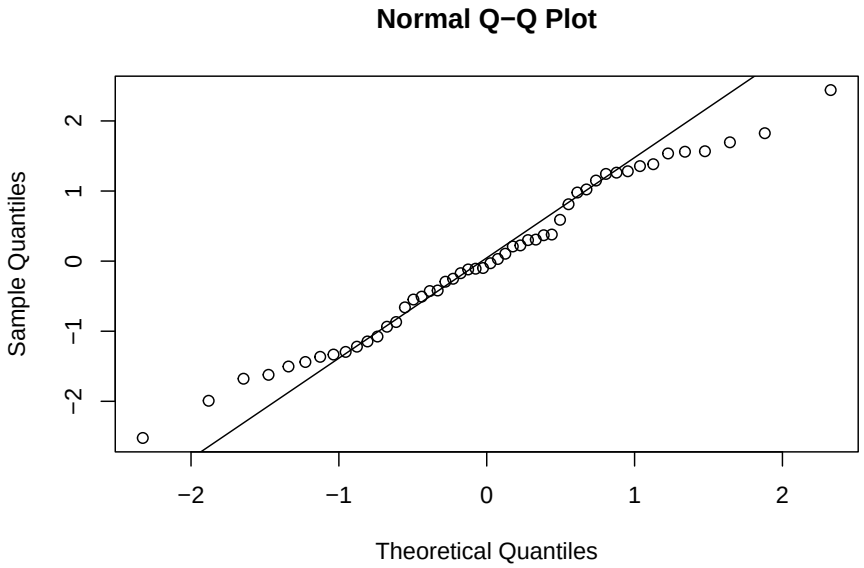



```
qqnorm(modelo)
```

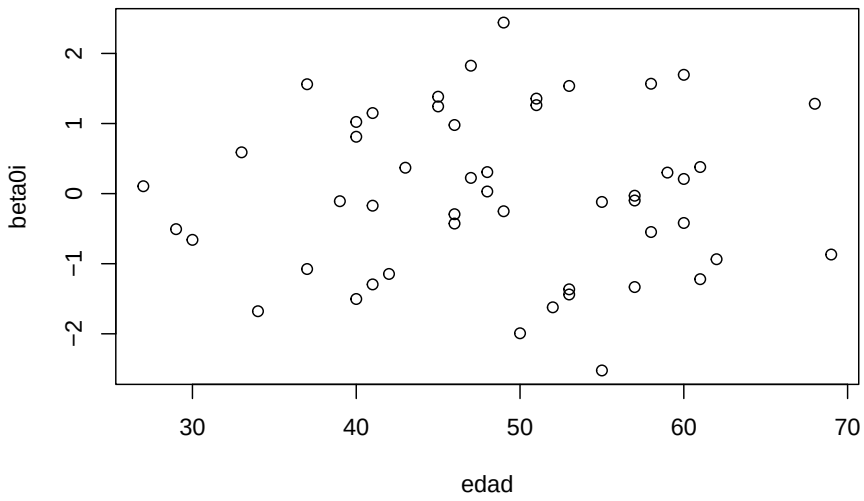


- Efectos aleatorios

```
beta0i<- ranef(modelo)[,1]
qqnorm(beta0i); qqline(beta0i)
```



```
edad <- with(datos, tapply(edad, indiv, mean))
# gráfico de los efectos aleatorios vs variables individuo
plot(edad, beta0i)
```



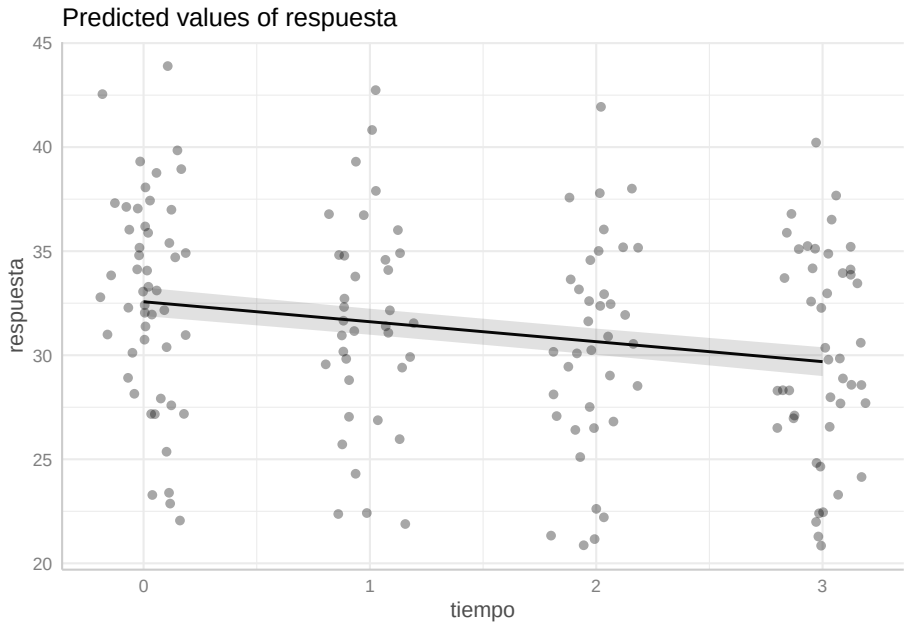
Parece que los efectos aleatorios siguen una distribución normal. Y no están relacionados con la edad.

Predicciones

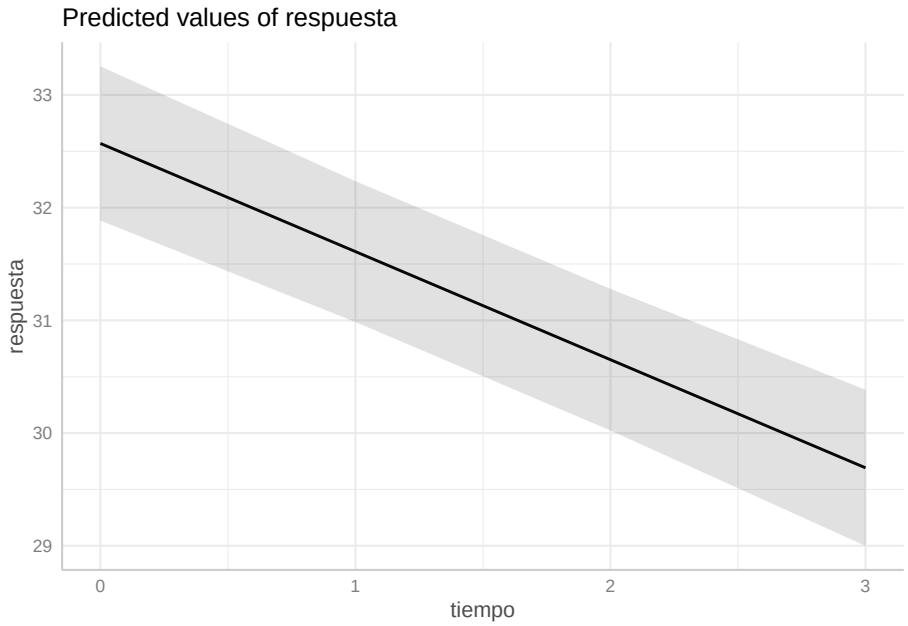
```
library(ggeffects)
pr <- ggpredict(modelo, terms = c("tiempo [all]"))
pr
```

```
# Predicted values of respuesta

tiempo | Predicted |         95% CI
-----------------------------------
     0 |     32.57 | [31.89, 33.25]
     1 |     31.61 | [30.99, 32.24]
     2 |     30.65 | [30.02, 31.28]
     3 |     29.69 | [29.00, 30.38]

Adjusted for:
*  edad = 48.00
* indiv = 26.00
```

```
plot(pr, add.data = TRUE)
```




```
plot(pr, residuals = TRUE)
```

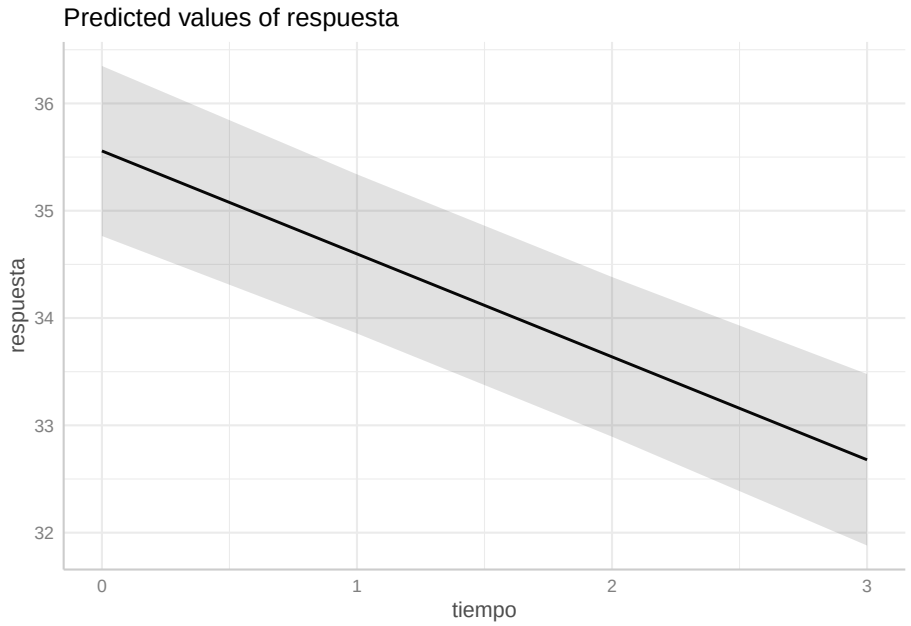


Las predicciones las realiza en la media de las covariables, en este caso la edad.
Si queremos que las predicciones las haga para un individuo de 55 años:

```
pr <- ggpredict(modelo, terms = c("tiempo [all]"), condition=c("edad"=55))
pr
```

```
# Predicted values of respuesta

tiempo | Predicted |         95% CI
-----------------------------------
     0 |     35.56 | [34.76, 36.35]
     1 |     34.60 | [33.86, 35.34]
     2 |     33.64 | [32.89, 34.38]
     3 |     32.68 | [31.88, 33.48]

Adjusted for:
* indiv = 26.00
```

```
plot(pr)
```



6.7.3.2 Comparación de los dos tratamientos

```
datos <- read.csv2("datos/imc.csv")
datos <- datos[order(datos$indiv,datos$tiempo),]
datos <- na.omit(datos) # para eliminar datos con missings (pero no individuos)
```

Volvemos al enunciado original trabajando con todos los datos. Ahora el objetivo es comparar la evolución de los dos tratamientos ajustando por la edad.

```
library(nlme)
modelo <- lme(respuesta ~ poly(tiempo, 3, raw=3) + tx:poly(tiempo, 3, raw=3) + edad,
              random= ~ poly(tiempo, 3, raw=3)  | indiv,
              data=datos,
              correlation=corCAR1(form = ~ tiempo|indiv))
summary(modelo)
```

```
Linear mixed-effects model fit by REML
  Data: datos
       AIC     BIC    logLik
  1417.874 1494.57 -688.9369

Random effects:
 Formula: ~poly(tiempo, 3, raw = 3) | indiv
 Structure: General positive-definite, Log-Cholesky parametrization
                         StdDev    Corr
(Intercept)              2.3069274 (Intr)
poly(tiempo, 3, raw = 3)1 1.9231468 -0.470
poly(tiempo, 3, raw = 3)2 1.7013762  0.354
poly(tiempo, 3, raw = 3)3 0.3562430 -0.342
Residual                 0.9432126

(Intercept)              p(,3,r=3)1 p(,3,r=3)2
poly(tiempo, 3, raw = 3)1
poly(tiempo, 3, raw = 3)2 -0.904
poly(tiempo, 3, raw = 3)3  0.852     -0.990
Residual

Correlation Structure: Continuous AR(1)
 Formula: ~tiempo | indiv
 Parameter estimate(s):
     Phi
0.248855
Fixed effects:  respuesta ~ poly(tiempo, 3, raw = 3) + tx:poly(tiempo, 3, raw = 3) +
                                Value Std.Error
(Intercept)                 11.168714 1.1301845
poly(tiempo, 3, raw = 3)1    2.402332 1.2555058
poly(tiempo, 3, raw = 3)2   -2.075890 1.1602056
poly(tiempo, 3, raw = 3)3    0.447207 0.2546185
edad                         0.442918 0.0224834
poly(tiempo, 3, raw = 3)1:tx -2.107063 0.7636948
poly(tiempo, 3, raw = 3)2:tx  0.905552 0.7097238
poly(tiempo, 3, raw = 3)3:tx -0.194165 0.1561157
```

```
                            DF   t-value
(Intercept)                244  9.882204
poly(tiempo, 3, raw = 3)1  244  1.913438
poly(tiempo, 3, raw = 3)2  244 -1.789243
poly(tiempo, 3, raw = 3)3  244  1.756379
edad                        98 19.699782
poly(tiempo, 3, raw = 3)1:tx 244 -2.759037
poly(tiempo, 3, raw = 3)2:tx 244  1.275922
poly(tiempo, 3, raw = 3)3:tx 244 -1.243724
                            p-value
(Intercept)                  0.0000
poly(tiempo, 3, raw = 3)1    0.0569
poly(tiempo, 3, raw = 3)2    0.0748
poly(tiempo, 3, raw = 3)3    0.0803
edad                         0.0000
poly(tiempo, 3, raw = 3)1:tx  0.0062
poly(tiempo, 3, raw = 3)2:tx  0.2032
poly(tiempo, 3, raw = 3)3:tx  0.2148
 Correlation:
                            (Intr) pl(,3,r=3)1
poly(tiempo, 3, raw = 3)1   -0.048
poly(tiempo, 3, raw = 3)2    0.033 -0.949
poly(tiempo, 3, raw = 3)3   -0.028  0.899
edad                        -0.975  0.027
poly(tiempo, 3, raw = 3)1:tx  0.020 -0.948
poly(tiempo, 3, raw = 3)2:tx -0.013  0.898
poly(tiempo, 3, raw = 3)3:tx  0.010 -0.849
                            pl(,3,r=3)2
poly(tiempo, 3, raw = 3)1
poly(tiempo, 3, raw = 3)2
poly(tiempo, 3, raw = 3)3   -0.989
edad                        -0.020
poly(tiempo, 3, raw = 3)1:tx  0.903
poly(tiempo, 3, raw = 3)2:tx -0.951
poly(tiempo, 3, raw = 3)3:tx  0.939
                            pl(,3,r=3)3 edad
poly(tiempo, 3, raw = 3)1
poly(tiempo, 3, raw = 3)2
poly(tiempo, 3, raw = 3)3
edad                         0.016
poly(tiempo, 3, raw = 3)1:tx -0.856      -0.020
poly(tiempo, 3, raw = 3)2:tx  0.941       0.013
poly(tiempo, 3, raw = 3)3:tx -0.951      -0.010
                            p(,3,r=3)1:
poly(tiempo, 3, raw = 3)1
poly(tiempo, 3, raw = 3)2
```

```
poly(tiempo, 3, raw = 3)3
edad
poly(tiempo, 3, raw = 3)1:tx
poly(tiempo, 3, raw = 3)2:tx -0.947
poly(tiempo, 3, raw = 3)3:tx  0.896
                            p(,3,r=3)2:
poly(tiempo, 3, raw = 3)1
poly(tiempo, 3, raw = 3)2
poly(tiempo, 3, raw = 3)3
edad
poly(tiempo, 3, raw = 3)1:tx
poly(tiempo, 3, raw = 3)2:tx
poly(tiempo, 3, raw = 3)3:tx -0.988

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3
-1.65153170 -0.36449401 -0.04684107  0.39784060
        Max
 1.80027927

Number of Observations: 350
Number of Groups: 100
```

Nota: `poly(tiempo, 3, raw=TRUE)` es lo miso que `tiempo + I(tiempo^2) + I(tiempo^3)`.

Fijate cómo se ha escrito la fórmula. De esta manera, cuando tiempo=0 (momento basal) no hay diferencias entre en los tratamientos.

Efectos aleatorios

```
anova(modelo, update(modelo, random = ~ 1 | indiv))
```

```
                                    Model df
modelo                                   1 20
update(modelo, random = ~1 | indiv)      2 11
                                          AIC
modelo                                 1417.874
update(modelo, random = ~1 | indiv) 1408.597
                                          BIC
modelo                                 1494.57
update(modelo, random = ~1 | indiv) 1450.78
                                       logLik
modelo                                 -688.9369
update(modelo, random = ~1 | indiv) -693.2985
                                        Test
modelo
update(modelo, random = ~1 | indiv) 1 vs 2
```

```
                                    L.Ratio
modelo
update(modelo, random = ~1 | indiv)  8.7233
                                    p-value
modelo
update(modelo, random = ~1 | indiv)  0.4632
```

Vemos en este caso, como el p-valor del LRT no coincide con la decisión basada en el AIC o el BIC. Podemos decantarnos con el modelo más simple, o sea, el que supone que los coeficientes del tiempo son fijos.

```
modelo <- update(modelo, random = ~ 1 | indiv)
```

Estructura de correlación de los errores

Comparamos con la matriz de independencia

```
anova(modelo, update(modelo, correlation=NULL))
```

```
                                   Model df
modelo                                 1 11
update(modelo, correlation = NULL)     2 10
                                        AIC
modelo                                1408.597
update(modelo, correlation = NULL) 1446.555
                                        BIC
modelo                                1450.780
update(modelo, correlation = NULL) 1484.903
                                      logLik
modelo                                -693.2985
update(modelo, correlation = NULL) -713.2774
                                      Test
modelo
update(modelo, correlation = NULL) 1 vs 2
                                     L.Ratio
modelo
update(modelo, correlation = NULL) 39.95779
                                    p-value
modelo
update(modelo, correlation = NULL)  <.0001
```

Nos quedamos con la estructura AR1.

Finalmente, comprovamos los efectos fijos:

```
coef(summary(modelo))
```

```
                                Value
(Intercept)                 10.9983674
poly(tiempo, 3, raw = 3)1    2.2104398
```

```
poly(tiempo, 3, raw = 3)2    -1.8762381
poly(tiempo, 3, raw = 3)3     0.4026454
edad                          0.4463927
poly(tiempo, 3, raw = 3)1:tx -1.9653182
poly(tiempo, 3, raw = 3)2:tx  0.7598944
poly(tiempo, 3, raw = 3)3:tx -0.1617992
                              Std.Error  DF
(Intercept)                  1.17012354 244
poly(tiempo, 3, raw = 3)1    1.28087236 244
poly(tiempo, 3, raw = 3)2    1.14921815 244
poly(tiempo, 3, raw = 3)3    0.25308582 244
edad                         0.02329682  98
poly(tiempo, 3, raw = 3)1:tx 0.78262641 244
poly(tiempo, 3, raw = 3)2:tx 0.70301744 244
poly(tiempo, 3, raw = 3)3:tx 0.15513365 244
                               t-value
(Intercept)                   9.399322
poly(tiempo, 3, raw = 3)1     1.725730
poly(tiempo, 3, raw = 3)2    -1.632621
poly(tiempo, 3, raw = 3)3     1.590944
edad                         19.161101
poly(tiempo, 3, raw = 3)1:tx -2.511183
poly(tiempo, 3, raw = 3)2:tx  1.080904
poly(tiempo, 3, raw = 3)3:tx -1.042967
                                 p-value
(Intercept)                  4.150278e-18
poly(tiempo, 3, raw = 3)1    8.566199e-02
poly(tiempo, 3, raw = 3)2    1.038384e-01
poly(tiempo, 3, raw = 3)3    1.129169e-01
edad                         6.509723e-35
poly(tiempo, 3, raw = 3)1:tx 1.268004e-02
poly(tiempo, 3, raw = 3)2:tx 2.808070e-01
poly(tiempo, 3, raw = 3)3:tx 2.979960e-01
```

Como era de esperar, la edad es muy significativa.

Contrastamos el efecto cuadrático y cúbico del tiempo (tanto para el grupo control como para el grupo de tratados):

```
modelo2 <- update(modelo, fixed = . ~ tiempo + tiempo:tx + edad)
coef(summary(modelo2))
```

```
                Value  Std.Error  DF    t-value
(Intercept) 11.0143619 1.16920806 248  9.4203610
tiempo       0.1947449 0.21775844 248  0.8943163
edad         0.4462810 0.02328366  98 19.1671311
tiempo:tx   -1.1381679 0.13574697 248 -8.3844810
```

```
                p-value
(Intercept) 3.281588e-18
tiempo      3.720198e-01
edad        6.352777e-35
tiempo:tx   3.860440e-15
```

```
anova(modelo, modelo2)
```

```
        Model df      AIC      BIC    logLik
modelo      1 11 1408.597 1450.780 -693.2985
modelo2     2  7 1394.327 1421.252 -690.1637
          Test  L.Ratio p-value
modelo
modelo2 1 vs 2 6.269749  0.1799
```

Como el p-valor del LRT es >0.05, nos quedamos con el modelo más simple, en que el tiempo tiene un efecto lineal en ambos grupos

```
modelo <- modelo2
coef(summary(modelo))
```

```
                Value  Std.Error  DF    t-value
(Intercept) 11.0143619 1.16920806 248  9.4203610
tiempo       0.1947449 0.21775844 248  0.8943163
edad         0.4462810 0.02328366  98 19.1671311
tiempo:tx   -1.1381679 0.13574697 248 -8.3844810
                p-value
(Intercept) 3.281588e-18
tiempo      3.720198e-01
edad        6.352777e-35
tiempo:tx   3.860440e-15
```

El efecto del tiempo en el grupo control no es significativo.

Para ver el efecto del tiempo en el grupo de tratamiento, cambiamos la categoria de referencia:

```
datos$tx <- factor(datos$tx, 2:1)
coef(summary(update(modelo)))
```

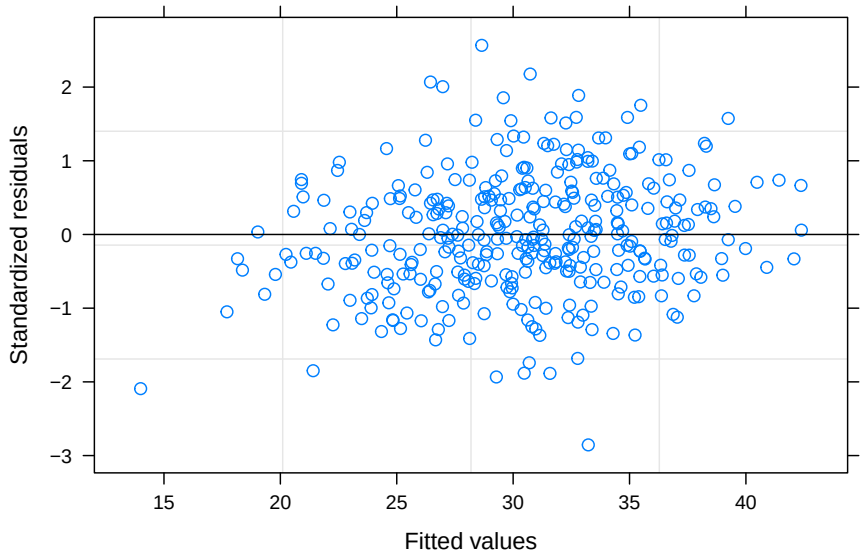
```
               Value  Std.Error  DF    t-value
(Intercept) 11.014362 1.16920806 248   9.420361
tiempo      -2.081591 0.10101027 248 -20.607715
edad         0.446281 0.02328366  98  19.167131
tiempo:tx1   1.138168 0.13574697 248   8.384481
                p-value
(Intercept) 3.281588e-18
tiempo      1.166972e-55
edad        6.352777e-35
```

```
tiempo:tx1  3.860440e-15
```

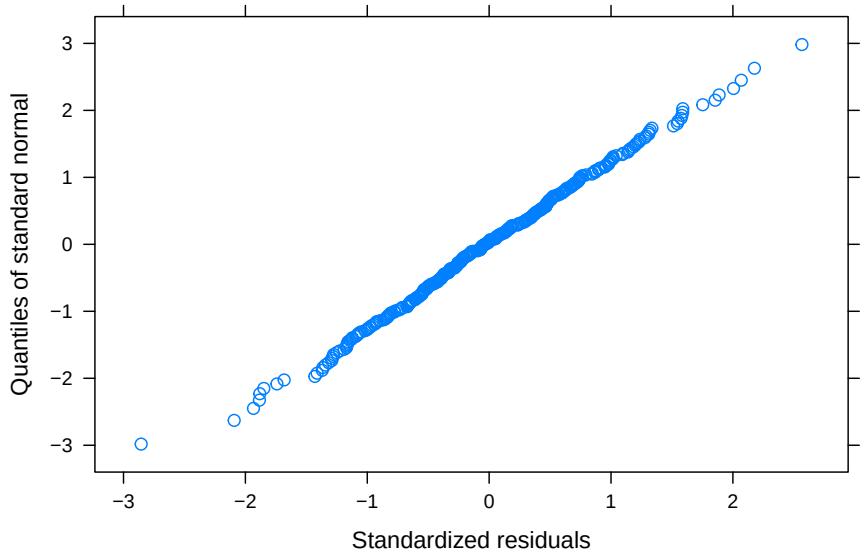
Vemos como el efecto del tiempo en el grupo de tratados es significativo y la pendiente es negativa.

Validación

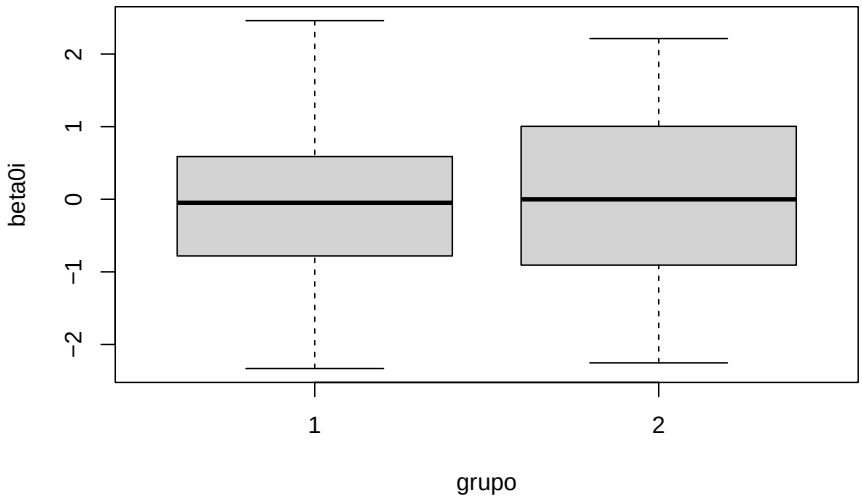
```
# residuos
plot(modelo)
```



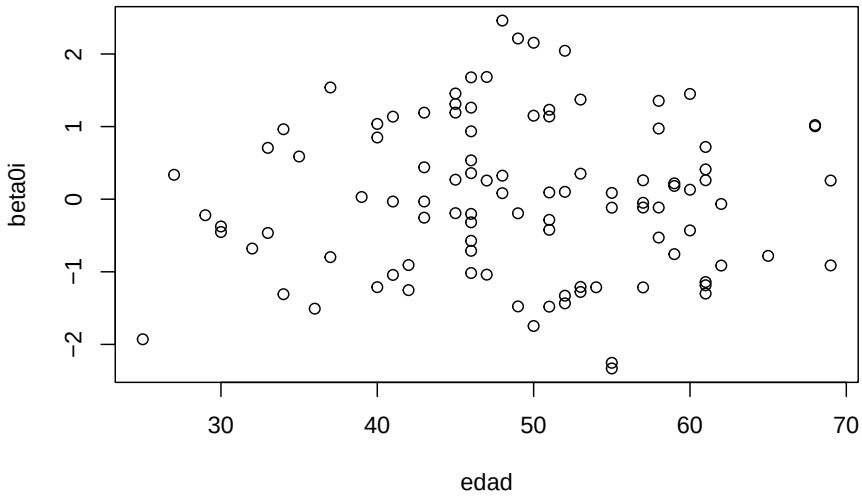

```
qqnorm(modelo)
```




```
# efectos aleatorios
beta0i <- ranef(modelo)[,1]
grupo <- with(datos, tapply(tx, indiv, head, n=1))
edad <- with(datos, tapply(edad, indiv, mean))
boxplot(beta0i ~ grupo)
```




```
plot(edad, beta0i)
```



Predicciones

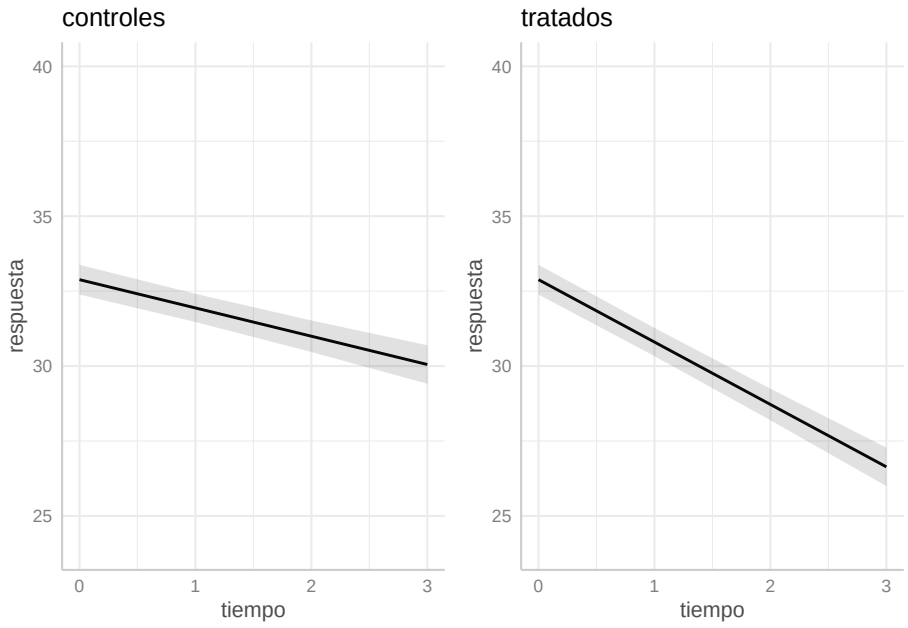
```
datos$tx <- factor(datos$tx, 1:2)
modelo <- update(modelo)

pr.cont <- ggpredict(modelo, terms = c("tiempo"), condition=c(tx=1), type="fixed")
pr.tx <- ggpredict(modelo, terms = c("tiempo"), condition=c(tx=2), type="fixed")
pr.cont
```

```
# Predicted values of respuesta

tiempo | Predicted |         95% CI
-----------------------------------
     0 |     32.88 | [32.39, 33.38]
     1 |     31.94 | [31.47, 32.41]
     2 |     31.00 | [30.47, 31.52]
     3 |     30.05 | [29.41, 30.69]

Adjusted for:
*  edad = 49.00
* indiv = 52.50
```

```
pr.tx
```

```
# Predicted values of respuesta

tiempo | Predicted |         95% CI
-----------------------------------
     0 |     32.88 | [32.39, 33.38]
     1 |     30.80 | [30.33, 31.27]
     2 |     28.72 | [28.19, 29.25]
     3 |     26.64 | [25.99, 27.28]

Adjusted for:
*  edad = 49.00
* indiv = 52.50
```

```
library(gridExtra)
grid.arrange(
    plot(pr.cont) + ylim(24,40) + ggtitle("controles"),
    plot(pr.tx) + ylim(24,40) + ggtitle("tratados"),
nrow=1, ncol=2)
```



6.8 Ejercicios

6.8.1 Ejercicio 4

Los datos `sleepstudy` de la librería `lme4` contienen información sobre 18 dindividuos que se han seguido durante 9 días, y para los cuales se ha registrado el tiempo de reacción (en milisegundos) tras haber estado en deprivación de sueño.

Los datos pueden cargarse mediante

```
library(lme4)
data(sleepstudy)
```

1. Crea un spaguetti plot para ver la evolución de la reacción a lo largo del tiempo
2. Modeliza el tiempo de reacción (variable respuesta) a lo largo de los días. Usa tanto polinomios como splines con la función `ns()`. Puedes usar la función `anova()` con el test LRT para contrastar si es necesario usar términos splines (`ns()`) o polinomios `poly()`).

6.8.2 Ejercicio 5

Los datos `dietox` de la librería `geepack` contienen el peso de cerdos medidos semanalmente durante 12 semanas cuando son sacrificados

(Weight). Los datos también contienen el peso en la semana 1 (Start), tres niveles diferentes de vitamina E (Evit - dosis: 0, 100, 200 mg dl-alfa-tocoferil acetato / kg de alimento) en combinación con 3 niveles diferentes de cobre (Cu - dosis: 0, 35, 175 mg / kg de alimento) que reciben en el alimento. También se registra la ingesta acumulada de alimento (Feed). Los cerdos son compañeros de camada (Litter). El objetivo principal es ver qué combinación de suplementos en el alimento hace que los cerdos alcancen un mayor peso.

1. Haz una descriptiva de las variables (summary)
2. ¿Qué variables son tiempo dependientes y cuáles no? (crea un gráfico para Weight, Evit y Cu)
3. Ajusta un modelo mixto para contestar a la pregunta científica
4. ¿Qué variables as considerado con efecto aleatorio?
5. ¿Qué estructura de correlaciones has considerado?
6. Intenta cambiar algunos aspectos del modelo (añadir quitar coeficientes aleatorios, estructura de correlación de los errores, simplificar o añadir variables/interacciones, …) para elegir el mejor modelo
7. Una vez escogido el modelo final, haz un gráfico de la evolución de la variable respuesta a diferentes niveles de `Start`.

Chapter 7

Análisis de supervivencia con datos longitudinales

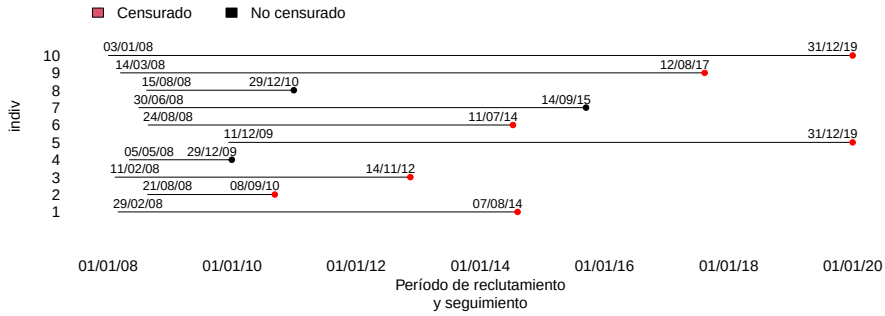
7.1 Tiempo hasta evento

En el análisis de superviviencia la variable respuesta es el **tiempo** hasta el evento de interés.

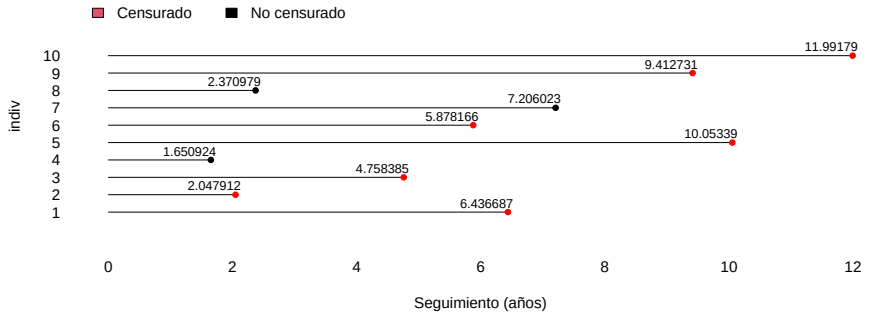
Normalente los datos se obtienen de un estudio de cohorte con seguimiento ya sea prospectivo o retrospectivo. Transcurrido el periodo de seguimiento o "follow-up time" puede que para alguno de los individuos de la muestra el evento de interés no se haya observado, ya sea porque ha finalizado el seguimiento o porque se han perdido o han tenido un evento diferente del de interés que ha interrumpido su seguimiento. En estos casos se suele decir que dichos individuos están **censurados**.

Es muy importante registrar el tiempo que ha pasado des del inicio hasta el evento para los no censurados y también el momento que se ha perdido el seguimiento para los censurados. Así hay que definir bien el momento de inicio y el momento final para cada participante del estudio. Y también es importante que el **mecanimso usado para obtener la información del seguimiento sea el mismo para todos.**

En la siguiente figura tenemos una descripción de cómo recogeríamos la información para diez individuos donde se observa que cada uno de ellos puede entrar en un momento distinto en el tiempo a partir del inicio del estudio (01/0108), que algunos se observa el evento de interés (puntos negros) y para otros el tiempo está censurado (puntos rojos) bien sea porque se acaba el periodo de seguimiento (01/01/2020) o porque abandonan el estudio antes del final (puntos rojos antes del 01/01/2020)



Normalment lo que hacemos es calcular el tiempo pasando toda la información a "tiempo cero". La siguiente figura muestra cómo quedarían los datos para el ejemplo anterior



De esta forma, para cada individuo anotaríamos la variable tiempo y crearíamos otra variable 0/1 que sería 0 para aquellos individuos censurados (puntos rojos) y 1 para los que observamos el evento de interés (puntos negros). En la práctica no se habla de variable censurada, si no de la variable evento, y es por eso que codificamos 0 a la censura y 1 a aquellos casos en los que observamos nuestro evento de interés.

7.1.1 Ejemplos

Pacientes diagnosticados de cancer de próstata. Seguimiento hasta recidiva o muerte. El inicio sería la fecha del diagnóstico y la fecha final sería la fechad e recidiva o muerte (para los no censurados) y la fecha de final de seguimiento para los censurados.

Estudio de una cohorte prospectiva a 10 años para estudiar el riesgo

de infarto agudo de miocardio incidente. La fecha de inicio sería la fecha de inclusión en el estudio, y la fecha final sería la fecha de ingreso por infarto o muerte por infarto (para los no censurados), y la fecha de final de seguimiento o fecha de muerte por otra causa (para los censurados).

7.1.2 Otros tipos de censura

La censura que se ha descrito es concretamente censura por la derecha. Esto quiere decir que cuando un dato está censurado significa que és superior al tiempo observado.

Existen otros tipo de censura que no estudiaremos:

- Censura por la izquierda: el tiempo es menor que el observado.
- Censura por intervalo: el tiempo se encuentra entre dos fechas o momentos determinados.
- Truncamiento por la izquierda: en realidad no es una censura, sino que es un retraso en el inicio del seguimiento. O sea, que el individuo lleva un tiempo en riesgo pero que ha entrado más tarde en el estudio.

7.2 Kaplan-Meier

El método de Kaplan-Meier se usa para estimar la supervivencia o su complementario, la probabilidad de que el evento ocurra antes del tiempo t.

Si no hubieran eventos censurados antes del tiempo t, la probabilidad de que ocurra el evento en este periodo es simplemente d_t/n donde d_t es el númerod e eventos antes de t y n el número de individuos de la cohorte. Pero qué pasa cuando un individuo está censurado antes de t? Lo contamos en el denominador o no? Ambas opciones dan resultados sesgados.

Kaplan-Meier propone un método para estimar el riesgo en cada momento t (o su supervivencia) que da resultados no sesgados ya que incorpora la información de los individos censurados hasta el momento que fueron seguidos.

Ejemplo

Analizaremos los datos `predimed` de la librería `compareGroups`. Se trata de una cohorte con tres grupos de intervención y con un seguimiento de unos 7 años. El evento de interés es el cardiovascular. En este caso, la variable tiempo está recogida en `toevent` y la variable que indica si un individuo está censurado es `event` que en este caso está codificada como `No` y `Yes`. Notemos que en este caso `No` correspondería a censura y `Yes` a no censura, pero que como hemos dicho anteriormente, nos interesa indentificar aquellos individuos cuyo tiemo corresponde al transcurrido hasta que ocurre el evento que estamos estudiando.

```
library(compareGroups)
data(predimed)
summary(predimed)
```

```
            group          sex
 Control       :2042   Male  :2679
 MedDiet + Nuts:2100   Female:3645
 MedDiet + VOO :2182


      age            smoke           bmi
 Min.   :49.00   Never  :3892   Min.   :19.64
 1st Qu.:62.00   Current: 858   1st Qu.:27.23
 Median :67.00   Former :1574   Median :29.76
 Mean   :67.01                  Mean   :29.97
 3rd Qu.:72.00                  3rd Qu.:32.46
 Max.   :87.00                  Max.   :51.94
     waist            wth          htn
 Min.   : 50.0   Min.   :0.3012   No :1089
 1st Qu.: 93.0   1st Qu.:0.5839   Yes:5235
 Median :100.0   Median :0.6258
 Mean   :100.4   Mean   :0.6283
 3rd Qu.:107.0   3rd Qu.:0.6687
 Max.   :177.0   Max.   :1.0000
  diab      hyperchol  famhist     hormo
 No :3322   No :1746   No :4895   No  :5564
 Yes:3002   Yes:4578   Yes:1429   Yes :  97
                                  NA's: 663


      p14            toevent        event
 Min.   : 0.000   Min.   :0.01643   No :6072
 1st Qu.: 7.000   1st Qu.:2.85832   Yes: 252
 Median : 9.000   Median :4.78850
 Mean   : 8.678   Mean   :4.35517
 3rd Qu.:10.000   3rd Qu.:5.79056
 Max.   :14.000   Max.   :6.99795
```

Para crear una variable censurada por la derecha se usa la función `Surv` del package `survival`.

```
library(survival)
```

Si la variable evento está codificada como 0/1 (0: censura 1:evento), como se

suele tener habitualmente, basta con escribir:

```
Surv(predimed$toevent, predimed$event)
```

En nuestro caso como la variable `event` es 'No' 'Yes', deberíamos indicar qué valor indica evento en la variable `event`

```
library(survival)
Surv(predimed$toevent, predimed$event=='Yes')[1:10]
```

```
 [1] 5.37440109  6.09719372+ 5.94661188+
 [4] 2.90759754  4.76112270+ 3.14852834
 [7] 0.71457905+ 4.90075302+ 0.04380561
[10] 0.88158798+
```

Notemos que se crea una nueva variable donde aquellos individuos censurados tiene un '+'

La función de supervivencia se puede estimar con el estimador de Kaplan-Meier mediante:

```
ss <- survfit(Surv(toevent, event=='Yes')~1, data=predimed)
```

Y podemos ver dichas estimaciones (para los primeros 6 tiempos de eventos) con la instrucción

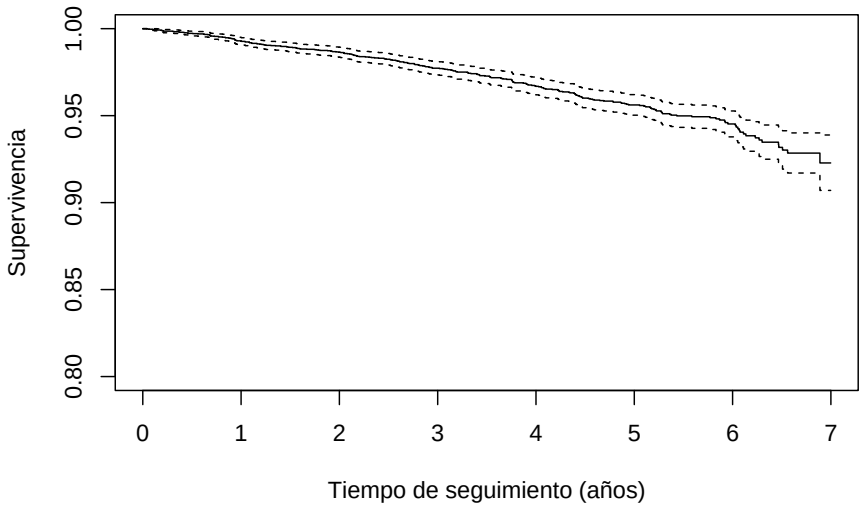
```
summary(ss, times=1:6)
```

```
Call: survfit(formula = Surv(toevent, event == "Yes") ~ 1, data = predimed)

 time n.risk n.event survival std.err
    1   6196      45    0.993 0.00106
    2   5602      39    0.986 0.00147
    3   4524      48    0.977 0.00196
    4   3723      44    0.967 0.00250
    5   2803      38    0.956 0.00301
    6   1116      23    0.945 0.00380
 lower 95% CI upper 95% CI
        0.991        0.995
        0.984        0.989
        0.973        0.981
        0.962        0.972
        0.950        0.962
        0.938        0.953
```

Normalmente lo que se suele hacer es visualizar las curvas de supervivencia mediante la instrucción

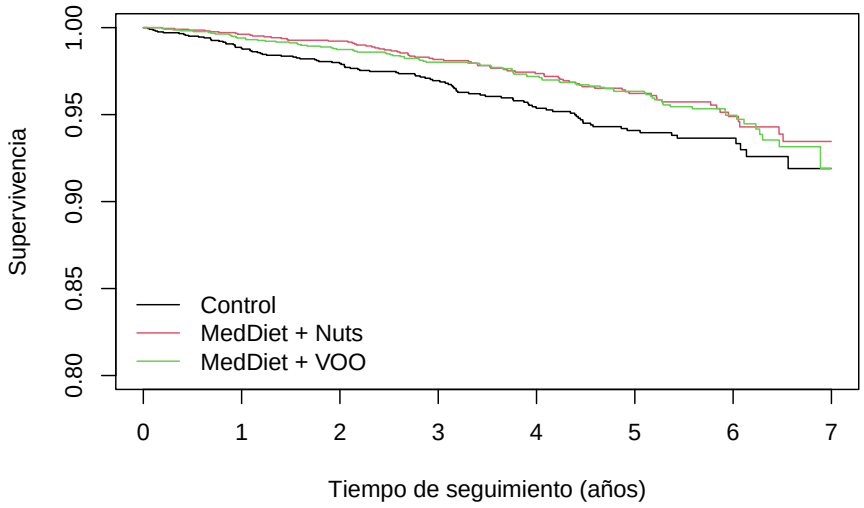
```
ans.km <- survfit(Surv(toevent, event=='Yes') ~ 1, predimed)
plot(ans.km, ylim=c(0.8,1),
```

```
     xlab="Tiempo de seguimiento (años)",
     ylab="Supervivencia")
```



Si quisiéramos calcular Kaplan-Meier para distrintos grupos, por ejemplo para el los distintos grupos de intervención de nuestro estudio, bastaría con escribir:

```
ans.km.group <- survfit(Surv(toevent, event=='Yes') ~ group, predimed)
plot(ans.km.group, ylim=c(0.8,1),
     xlab="Tiempo de seguimiento (años)",
     ylab="Supervivencia", col=1:3)
legend("bottomleft", levels(predimed$group),
       lty=1, col=1:3, bty="n")
```



Finalmente, podemos comparar las curvas de supervivencia entre grupos con la función `survdiff` que tiene implementado por defecto, el test de log-rank:

```
survdiff(Surv(toevent, event=='Yes') ~ group, predimed)
```

```
Call:
survdiff(formula = Surv(toevent, event == "Yes") ~ group, data = predimed)

                       N Observed Expected
group=Control       2042       97     75.4
group=MedDiet + Nuts 2100       70     82.7
group=MedDiet + VOO  2182       85     93.9
                     (O-E)^2/E (O-E)^2/V
group=Control            6.194      8.85
group=MedDiet + Nuts     1.946      2.90
group=MedDiet + VOO      0.848      1.35

 Chisq= 9  on 2 degrees of freedom, p= 0.01
```

Podemos concluir que las diferencias observadas en las curvas de supervivencia, son significativamente distintas ya que el p-valor del test de log-rank es $<0.5\%$.

Este test considera que todas las diferencias observadas a lo largo del tiempo son igual de imporatantes. A veces, queremos dar más peso a las diferencias observadas al inicio del estudio. En ese caso, el test más potente es el del Wilcoxon que puede calcularse de la misma manera, pero usando el argumento `rho=1`

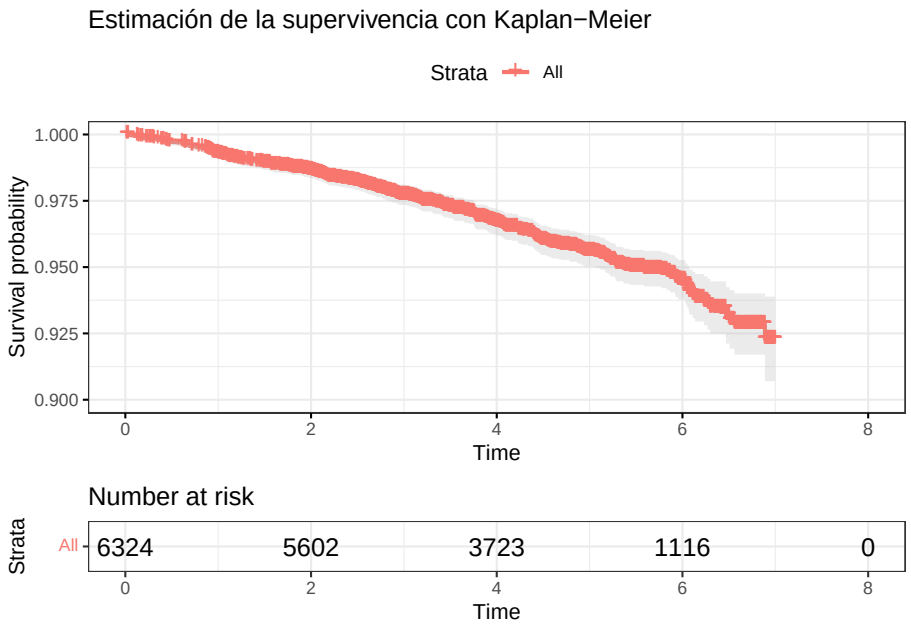
```
survdiff(Surv(toevent, event=='Yes') ~ group, predimed, rho = 1)
```

```
Call:
survdiff(formula = Surv(toevent, event == "Yes") ~ group, data = predimed,
    rho = 1)

                       N Observed Expected
group=Control       2042     95.0     73.6
group=MedDiet + Nuts 2100     68.1     80.7
group=MedDiet + VOO  2182     82.7     91.6
                    (O-E)^2/E (O-E)^2/V
group=Control           6.222      9.11
group=MedDiet + Nuts    1.952      2.98
group=MedDiet + VOO     0.857      1.40

 Chisq= 9.3  on 2 degrees of freedom, p= 0.01
```

Llegamos a la misma conclusión que con el test de log-rank, pero notemos que el valor del estadistico (Chisq) es ligeramente superior, por lo que el p-valor es menor (es decir, más significativo) y nos daría más evidencias en contra de la hipótesis nula (notemos que aquí vemos 0.01 en ambos casos por un tema de redondeo).

Podemos mejorar la visualización usando la función `ggsurvplot ()` de la librería `survminer`. Una caída vertical en las curvas indica un evento. Una marca vertical en las curvas significa que un individuo fue censurado.

```
library(survminer)
ggsurvplot(
  ans.km, ylim=c(0.9,1),
  pval = TRUE,
  conf.int = TRUE,
  risk.table = TRUE,
  ggtheme = theme_bw(),
  title = "Estimación de la supervivencia con Kaplan-Meier"
)
```



NOTA: la opción `pval=TRUE` nos permitiría ver el p-valor de Kaplan-Meier en el gráfico, pero puesto que hemos indicado que el eje Y se vea sólo de 0.9 a 1, el p-valor no se ve. Si se vería en caso de quitar la opción de `ylim` aunque entonces las curvas se verían muy juntas. Existen opciones para poder "tunear" esta visualización usando la función `annotate()`.

7.3 Funciones involucradas en el análisis de supervivencia

Aparte de la función de supervivencia que se define como:

- **Supervivencia:** probabilidad de estar libre de evento en el momento t (se supone que el evento ocurre después)

$$S(t) = \Pr(T > t)$$

Existen otras medidas para resumir este tipo de estudios que pueden ser interesantes según el contexto. Por ejemplo, si nos interesa cuantificar la probabilidad de observar nuestro evento de interés (normalemete cuanod el evento no es "malo" como en el análisis de supervivencia tradicional que el evento es la muerte) podemos calcular la función de:

- **Distribución**: probabilidad de evento antes de tiempo t. Es el complementario de la función de supervivencia

$$\Pr(T \leq t) = 1 - S(t)$$

Otras medidas interesantes son:

- **Hazard** (riesgo instantaneo): Es la probabilidad que ocurra el evento en un intervalo infinitamente pequeño dado que no lo ha tenido hasta el momento t

$$\lambda(t) = \lim_{\delta \to 0} \frac{\Pr\left(T \in (t, t+\delta)\right)}{S(t)}$$

- **Cumulative Hazard** (riesgo acumulado): es la suma o integral del riesgo instantáneo hasta el momento t

$$\Lambda(t) = \int_0^t \lambda(s) ds$$

Existe la siguiente relación entre el riesgo acumulado y la función de supervivencia

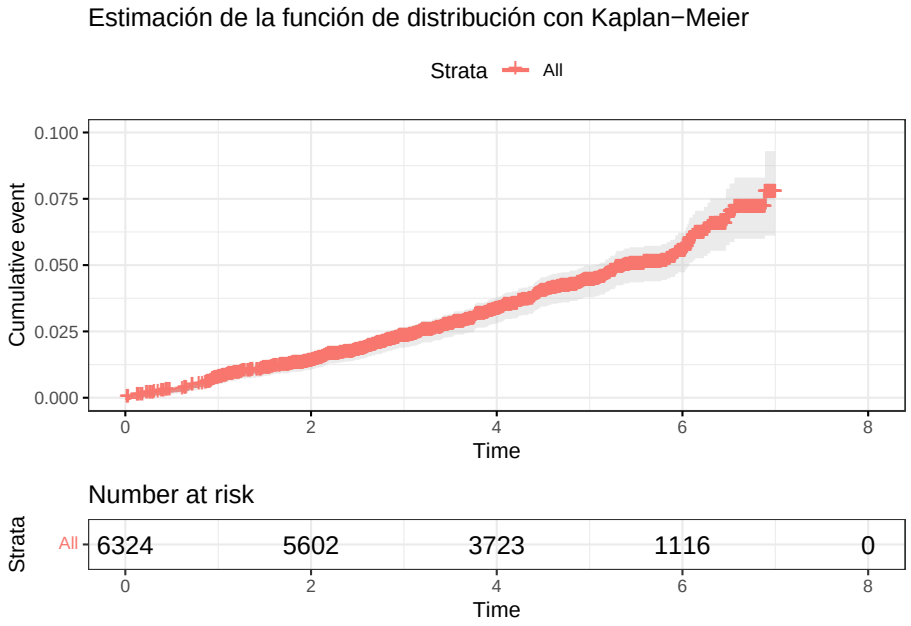
$$S(t) = \exp\left(-\Lambda(t)\right)$$

o bien

$$\Lambda(t) = -\ln\left(S(t)\right)$$

Todas estas funciones se pueden calcular conla función `ggsurvplot()` cambiando el argumento `fun`. Por ejemplo la función de probabilidad se calcularía mediante la opción "event" y la de riesgo acumulado con "cumhaz"

```
ggsurvplot(
  ans.km, ylim=c(0,.1),
  fun = "event",
  conf.int = TRUE,
  risk.table = TRUE,
  ggtheme = theme_bw(),
  title = "Estimación de la función de distribución con Kaplan-Meier"
)
```



7.4 Modelo de regresión de Cox

Normalmente queremos estudiar cómo influye más de una variable en la supervivencia. Para este caso, necesitamos utilizar modelos de regresión. Los modelos de regresión de Cox sirven para evaluar el efecto de distintas variables sobre el tiempo hasta el evento, o para crear un moelo de predicción.

Los modelos de cox asumen **riesgos proporcionales**, esto es, se separa el riesgo ("Hazard") de paceder un evento antes del momento t como un producto del

- $\Lambda_0(t)$: riesgo basal, cuando todas las variables independientes x valen cero) y
- $\sum_{k=1} \beta_k x_{ik}$: combinación lineal de las variables independientes (predictor lineal).

$$\Lambda(t|\vec{x}_i) = \Lambda_0(t) \cdot \exp\left(\sum_{k=1} \beta_k x_{ik}\right)$$

donde los coeficientes β_k son los log-Hazard Ratios.

Cox propone un método para estimar los coeficientes β_k sin suponer ninguna distribución sobre la variable respuesta T (tiempo hasta evento). Por esto se llama método semiparamétrico y se basa en estimar la "partial-likelihood".

Existen otros métodos que suponen una distribución de la T, y por lo tanto parametrizan la incidencia basal $\Lambda_0(t)$. Por ejemplo,la regresión de Weibull que supone una distribución Weibull sobre T. Una de las ventaja que tienen los métodos no paramétricos es que permiten estimar la media o la mediana de T aunque más de la mitad de los individuos de la muestra estén censurados (o sea, que no se llegue al 50% de eventos en el seguimiento). La desventaja es que suponen una distribución sobre T que puede no ser correcta y que conllevaría a resultados sesgados. En biomedicina, el método más usado es el de los modelos de Cox y es el que estudiaremos en este curso.

Ejemplo

Para ajustar un modelo de Cox en R se usa la función `coxph` de la librería `survival`.

```
modelo <- coxph(Surv(toevent, event=='Yes')~age+sex+p14+group, predimed)
```

Hay diferentes aspectos a validar del modelo de Cox. Entre ellos la proporcionalidad de los efectos. Quiere decir que se supone que las β_k no dependen del tiempo (por ejemplo, el efecto del sexo es el mismo tanto a 1 años como a 5 años). Ésto se puede comprovar mediante la siguiente función:

```
cox.zph(modelo)
```

```
         chisq df     p
age    0.33102  1 0.565
sex    0.17845  1 0.673
p14    0.00189  1 0.965
group  5.75156  2 0.056
GLOBAL 6.35652  5 0.273
```

Aparece un p-valor para cada variable y uno global. En este caso parece que se cumple la proporcionalidad para todas las variables ya que el p-valor no es < 0.05 y por lo tanto no podemos rechazar la hipótesis nula que es que los riesgos son proporcionales. No obstante, si no se cumpliera la proporcionalidad de una variable categórica, por ejemplo el sexo, ésta se puede poner como `strata` (se asume una curva de indidencia basal $\Lambda_0(t)$ para cada sexo) y se solucionaría el problema. Cuando esto no ocurre para una variable continua, debemos hacer modelos más avanzados que contemplan la posibilidad de introducir en el modelo una variable dependiente del tiempo (que veremos más adelante).

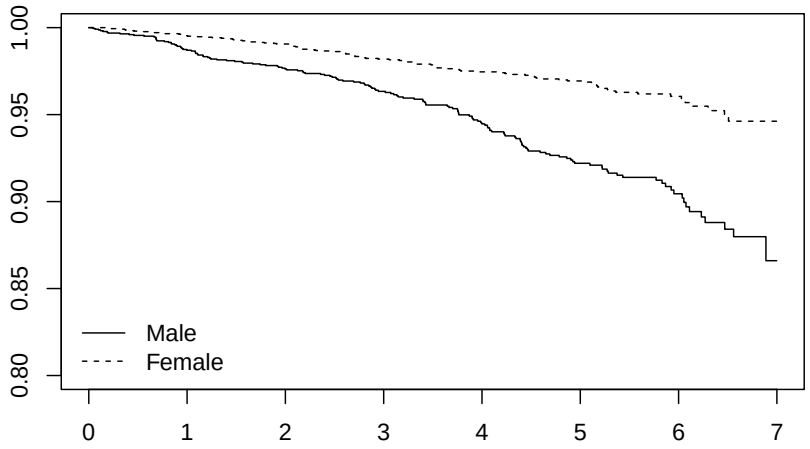
```
modelo2 <- coxph(Surv(toevent, event=='Yes')~age+strata(sex)+p14+group, predimed)
summary(modelo2)
```

```
Call:
coxph(formula = Surv(toevent, event == "Yes") ~ age + strata(sex) +
    p14 + group, data = predimed)

  n= 6324, number of events= 252
```

```
                        coef exp(coef) se(coef)
age                  0.06790   1.07026  0.01010
p14                 -0.12221   0.88497  0.03046
groupMedDiet + Nuts -0.39498   0.67369  0.15771
groupMedDiet + VOO  -0.31459   0.73009  0.14894
                          z Pr(>|z|)
age                   6.720 1.82e-11 ***
p14                  -4.012 6.03e-05 ***
groupMedDiet + Nuts -2.505   0.0123 *
groupMedDiet + VOO  -2.112   0.0347 *
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

                    exp(coef) exp(-coef)
age                    1.0703     0.9344
p14                    0.8850     1.1300
groupMedDiet + Nuts    0.6737     1.4844
groupMedDiet + VOO     0.7301     1.3697
                    lower .95 upper .95
age                    1.0493    1.0917
p14                    0.8337    0.9394
groupMedDiet + Nuts    0.4946    0.9177
groupMedDiet + VOO     0.5453    0.9776

Concordance= 0.652  (se = 0.018 )
Likelihood ratio test= 71.75  on 4 df,   p=1e-14
Wald test            = 72.99  on 4 df,   p=5e-15
Score (logrank) test = 73.94  on 4 df,   p=3e-15
```

En este gráfico se obtiene una curva de supervivencia para cada sexo, ajustado por las demás covariables (nota que en esta gráfica se asume que el efecto de las demás covariables es el mismo para ambos sexos).

```
plot(survfit(modelo2), ylim=c(0.8,1), lty=1:2)
legend("bottomleft", levels(predimed$sex), lty=1:2, bty="n")
```



7.4.1 Efectos tiempo-dependientes

El efecto tiempo-dependiente (no confundir con variables tiempo-dependientes) se da cuando el efecto de una variable β_k no es constante a lo largo del tiempo. En este caso, como se ha comentado anteriormente, si se trata de una variable categórica se puede poner en el modelo como `strata`. Si se trata de una variable continua, se puede incorporar la interacción de la variable x_k con el tiempo. Otra estrategia que se usa habitualmente es dividir el tiempo de seguimiento en dos o tres tramos (a corto y a largo plazo), y realizar análisis por separado.

De la anterior ecuación sobre la incidencia acumulada, se suponía que los efectos eran fijos y no dependían del tiempo. Pero si dependieran del tiempo en general se debería reescribir como:

$$\Lambda(t|\vec{x}_i) = \Lambda_0(t) \cdot \exp\left(\sum_{k=1} \beta_k(t) x_{ik}\right)$$

donde $\beta_k(t)$ representa una función del tiempo.

Este tipo de modelos con efectos tiempo dependientes no los veremos en este curso, sin embargo nos centraremos en otro aspecto fundamental que se da en estudios longitudinales que es el hecho de recoger una variable explicativa en distintos momentos del tiempo (variales tiempo-dependientes).

7.4.2 Variables tiempo-dependientes (datos longitudinales)

Los modelos con variables tiempo-dependientes se tienen cuando en el seguimiento de los individuos de la muestra también se han ido actualizando los valores de todas o algunas de las variables indepndientes (x_k). Por ejemplo, el nivel de colesterol se puede recoger al inicio del estudio e introducir esa variable en el modelo de Cox para ver si influencia en el tiempo hasta sufrir un infarto de miocardio, pero también podemos recoger el nivel de colesterol en distintos momentos temporales y ver si esta variable cambiante a lo largo del tiempo se asocia con nuestro evento de interés.

Así pues, en cada momento t el riesgo acumulado se tiene que estimar teniendo en cuenta el valor que toma cada x_k en dicho tiempo t. Esto se puede formular de la siguiente manera

$$\Lambda(t|\vec{x}_i = \Lambda_0(t) \cdot \exp\left(\sum_{k=1} \beta_k x_{itk}\right)$$

donde x_{itk} representa el valor de la variable x_k del individuo i en el momento t

7.4.3 Estructura de los datos

Para ajustar estos modelos, el reto principal (y único), es estructurar bien la base de datos. Así, para cada individuo tendremos tantas filas como actualizaciones tengamos de cada variable x_k. Además hay que anotar el momento de estos cambios. Adicionalemente tendremos una fila final donde se indicará el tiempo de evento o censura para nuestro evento de interés.

Veámoslo con un ejemplo: Utilizaremos la base de datos `aids` de la librería `JM`

```
library(JM)
data(aids)
head(aids)
```

```
  patient  Time death       CD4 obstime drug
1       1 16.97     0 10.677078       0  ddC
2       1 16.97     0  8.426150       6  ddC
3       1 16.97     0  9.433981      12  ddC
4       2 19.00     0  6.324555       0  ddI
5       2 19.00     0  8.124038       6  ddI
6       2 19.00     0  4.582576      12  ddI
  gender prevOI          AZT start  stop event
1   male   AIDS intolerance     0  6.00     0
2   male   AIDS intolerance     6 12.00     0
3   male   AIDS intolerance    12 16.97     0
4   male noAIDS intolerance     0  6.00     0
5   male noAIDS intolerance     6 12.00     0
```

```
6   male noAIDS intolerance    12 18.00     0
```

En esta base de datos tenemos diferentes participantes en los que se ha tomado distintas medidas de la variable CD4. La variable `obstime` indica cuándo se han tomado las medidas de CD4. Mientras que la variable `time` y `death` indica el tiempo observado y si el individuo se ha muerto (1) o sigue vivo (0, dato censurado) al finalizar el seguimiento. En este caso el evento de interés es la muerte y los individuos vivos serán los censurados. La variable tiempo-dependiente es la variable CD4. Nuestro objetivo final es demostrar si hay diferencias en la mortalidad entre dos fármacos (variable 'drug': ddI = didanosine; ddC = zalcitabine.) ajustando por la variable 'gender'.

Para ajustar un modelo con variables tiempo-dependientes se ha de reestructurar esta base de datos. Para ello debemos llevar a cabo los siguientes pasos

1. Creamos una base de datos con una fila por individuo, con los tiempos de muerte y las covariables de interés (fijas, no cambiantes a lo largo del tiempo - en nuestro caso 'drug' , 'gender') y creamos la variable `endpt`.

```
temp <- aids %>% dplyr::select(patient, Time, death, drug, gender)
x <- rep(1,nrow(temp))
datos <- aggregate(x, temp, sum)
datos <- tmerge(datos, datos, id=patient, endpt = event(Time, death))
head(datos)
```

```
  patient  Time death drug gender x tstart tstop
1     351 12.27     0  ddC female 4      0 12.27
2     305 12.30     0  ddC female 2      0 12.30
3     336 12.57     0  ddC female 3      0 12.57
4     268 12.73     0  ddC female 4      0 12.73
5     160 13.20     0  ddC female 3      0 13.20
6     377 13.50     0  ddC female 4      0 13.50
  endpt
1     0
2     0
3     0
4     0
5     0
6     0
```

2. Luego, hacemos uso de la función `tmerge()` para crear la base de datos en el formato deseado. Las variables tiempo dependientes se especifican mediante la función `tdc` en que se indica también la variable que recoge cuando se han tomado sus medidas (en nuestro caso 'obstime').

```
aids2 <- tmerge(datos, aids, id=patient, CD4 = tdc(obstime, CD4))
head(aids2)
```

```
  patient  Time death drug gender x tstart tstop
```

```
1     351 12.27     0  ddC female 4      0  2.00
2     351 12.27     0  ddC female 4      2  6.00
3     351 12.27     0  ddC female 4      6 12.00
4     351 12.27     0  ddC female 4     12 12.27
5     305 12.30     0  ddC female 2      0  6.00
6     305 12.30     0  ddC female 2      6 12.30
  endpt      CD4
1     0 5.477226
2     0 6.403124
3     0 4.690416
4     0 4.000000
5     0 2.000000
6     0 2.449490
```

7.4.4 Ajuste del modelo

En esta nueva base de datos, tenemos intervalos de tiempo `tstart` y `tstop` que se usará como tiempos de supervivencia en la función `Surv` y que ayuda a dividir el seguimento en los intervalos donde CD4 has sido observado de forma diferente para cada invididuo. Con esta información, podremos usar la función `coxph ()` de la forma habitual, pero usando esta nueva escala de tiempo:

```
modelo <- coxph(Surv(tstart, tstop, endpt) ~ CD4 + drug + gender, data=aids2)
summary(modelo)
```

```
Call:
coxph(formula = Surv(tstart, tstop, endpt) ~ CD4 + drug + gender,
    data = aids2)

  n= 1405, number of events= 188

               coef exp(coef) se(coef)      z
CD4        -0.19440   0.82333  0.02434 -7.986
drugddI     0.31698   1.37298  0.14669  2.161
gendermale -0.32579   0.72196  0.24248 -1.344
           Pr(>|z|)
CD4         1.4e-15 ***
drugddI      0.0307 *
gendermale   0.1791
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

           exp(coef) exp(-coef) lower .95
CD4           0.8233     1.2146    0.7850
drugddI       1.3730     0.7283    1.0299
```

```
gendermale    0.7220     1.3851    0.4489
           upper .95
CD4           0.8636
drugddI       1.8303
gendermale    1.1612

Concordance= 0.697  (se = 0.018 )
Likelihood ratio test= 96.28  on 3 df,   p=<2e-16
Wald test            = 67.58  on 3 df,   p=1e-14
Score (logrank) test = 75.22  on 3 df,   p=3e-16
```

Notemos que que tanto las variables tiempo-dependientes (en este caso CD4) como las no-tiempo-dependientes (`drug` y `gender`) se ponen de la misma manera y de forma habitual en la fórmula.

La interpretación de los resultados es exactamente la misma que para un modelo de Cox sin variables tiempo-dependientes.

Comparación con el modelo sin variables tiempo-dependientes

Estos resultados los podríamos comparar con el caso en el que consideráramos la primera medida de CD4 como una variable fija a lo largo del tiempo:

```
aids1obs <- subset(aids, obstime==0)
modelo1obs <- coxph(Surv(Time, death) ~ CD4 + drug + gender, aids1obs)
summary(modelo1obs)
```

```
Call:
coxph(formula = Surv(Time, death) ~ CD4 + drug + gender, data = aids1obs)

  n= 467, number of events= 188

               coef exp(coef) se(coef)      z
CD4        -0.18295   0.83281  0.02222 -8.232
drugddI     0.26694   1.30597  0.14648  1.822
gendermale -0.20238   0.81679  0.24216 -0.836
           Pr(>|z|)
CD4          <2e-16 ***
drugddI      0.0684 .
gendermale   0.4033
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

           exp(coef) exp(-coef) lower .95
CD4           0.8328     1.2008    0.7973
drugddI       1.3060     0.7657    0.9801
gendermale    0.8168     1.2243    0.5081
```

```
           upper .95
CD4           0.8699
drugddI       1.7403
gendermale    1.3129

Concordance= 0.703  (se = 0.019 )
Likelihood ratio test= 92.54  on 3 df,   p=<2e-16
Wald test            = 70.82  on 3 df,   p=3e-15
Score (logrank) test = 78.19  on 3 df,   p=<2e-16
```

Notemos que utilizando este modelo, nuestra conclusión sería que no hay diferencias en la supervivencia entre fármacos ('drug') mientras que el modelo utilizando datos longitudinales para CD4 muestra que la didanosine (ddI) tiene una peor supervivencia (p = 0.0307).

Una vez más se demuestra que, en estadística, uno puede usar cualquier modelo para analizar sus datos y obtener resultados similares (en este segundo modelo casi sale significativo). Sin embargo, si no se utiliza el modelo correcto el perjudicado es el investigador, ya que, la utilización del modelo correcto proporciona el test más potente para detectar diferencias cuando realmente las hay. Es como el caso de analizar una variable 0,1 para comparar dos grupos y usar la t de Student. R nos dará un p-valor, pero este no será el test más potente para encontrar diferencias cuando realmente las haya, ya que ese test es el más potente cuando los datos son normales. Es por ello que en estos casos se usa la chi-cuadrado.

Validación del modelo

La validación del modelo con variables tiempo-dependientes se hace de la misma manera que para el modelo de Cox "clásico". Por ejemplo, también se puede aplicar la función `coxz.ph`. Sin embargo, la discriminación y la calibración que necesitan del cálculo del riesgo predicho a tiempo t_0 no és fácil de calcular: ¿cómo se tiene en cuenta que el valor de x_k cambia y que ello conlleva a un cambio del riesgo acumulado Λ?.

```
cox.zph(modelo)
```

```
        chisq df    p
CD4    0.2875  1 0.59
drug   0.0034  1 0.95
gender 1.6437  1 0.20
GLOBAL 1.9692  3 0.58
```

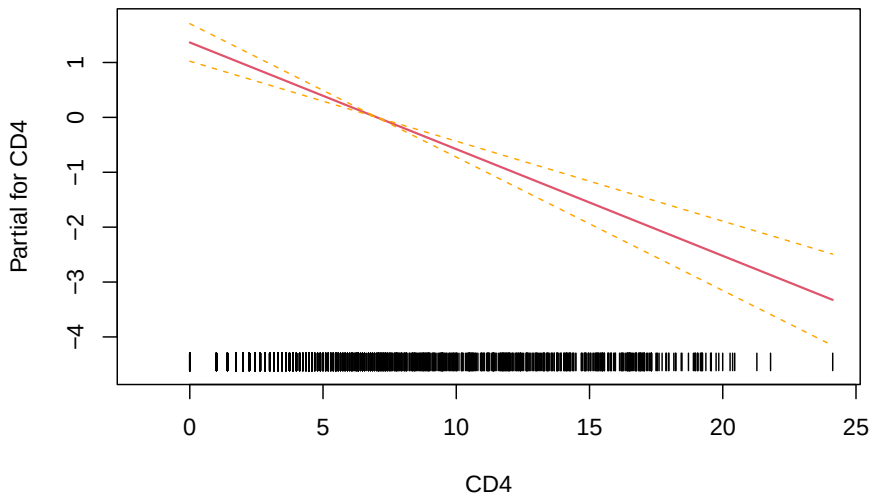
Términos no lineales: "splines"

También, como en los modelo de Cox "clásicos" se pueden introducir términos polinómicos o de splines (`psplines`) para modelar efectos no lineases de las variables x_k cuantitativas.

```
modelo.splines <- coxph(Surv(tstart, tstop, endpt) ~ pspline(CD4) + drug + gender, data
coef(summary(modelo.splines))
```

```
                           coef   se(coef)
pspline(CD4), linear -0.1900175 0.02553879
pspline(CD4), nonlin         NA         NA
drugddI               0.3164310 0.14684398
gendermale           -0.3282002 0.24253118
                           se2     Chisq
pspline(CD4), linear 0.02527049 55.358799
pspline(CD4), nonlin         NA  5.669855
drugddI              0.14682270  4.643503
gendermale           0.24249275  1.831229
                           DF            p
pspline(CD4), linear 1.000000 1.004194e-13
pspline(CD4), nonlin 3.092747 1.367478e-01
drugddI              1.000000 3.117131e-02
gendermale           1.000000 1.759825e-01
```

```
termplot(modelo, terms=1, se=TRUE, rug=TRUE)
```



en nuestro ejemplo, se demuestra que el efecto de CD4 es lineal (no hay términos cuadráticos, ni cúbicos, ni puntos de inflexión o cambios de tendencias, ...) por lo que el modelo sin splines ya sería suficiente para modelar nuestros datos.

Chapter 8

Joint models para datos longitudinales y datos de supervivencia

En el capítulo anterior hemos visto cómo modelar una variable dependiente que mide el tiempo hasta una variable de interés (análisis de supervivencia). Vimos también cómo tener en cuenta como estimar modelos de supervivencia cuando una variable independiente es medida a lo largo del tiempo. En este tema, iremos un paso más allá y nos interesaremos por unos modelos en los que nuestra variable resultado está formada por dos tipos de variables: una variable respuesta medida de forma longitudinal y otra que recoge el tiempo hasta un evento de interés. A estos modelo se les comonce como **"Joint Models"**. Estos modelos tienen en cuenta las variables tiempo-dependientes para modelizar el tiempo hasta evento con posible censura por la derecha. Estos modelos surgen cuando hay valores faltantes en algunas medidas de las variables x_k. Para solventarlo, modelizan una regresión de supervivencia de Cox, y en lugar de condicionar por los valores observados de x_k, se condiciona por los valores ajustados de ellos según modelos de datos longitudinales que pueden ser los modelos mixtos. De esta forma se **fusiona los modelos de medidas repetidas con los modelos de Cox.**.

En este curso mostraremos cómo llevar a cabo estos análisis mediante la librería `JM`, pero debemos tener en cuenta que también podemos usar la librería `joineR` para ajustar este tipo de modelos. Esta librería también tiene incorporadas funciones y opciones para tener encuenta **eventos competitivos**, es decir, cuando nuestro evento de interés no es uno sólo si no más de uno.

8.1 ¿Por qué deberíamos utilizar este tipo de modelos?

Como mencionamos en la sección anterior, el modelo de riesgos proporcionales de Cox se puede ampliar para incorporar variables dependientes del tiempo. Sin embargo, cuando enfocamos nuestro interés en el tiempo hasta el evento y deseamos tener en cuenta el efecto de la variable longitudinal como una covariable dependiente del tiempo, los enfoques tradicionales para analizar los datos del tiempo hasta el evento (como usar la verosimilitud parcial para los modelos de Cox) no son aplicables en todas las situaciones.

En particular, los modelos estándar de tiempo hasta el evento requieren que las covariables dependientes del tiempo sean externas; es decir, el valor de esta covariable en el momento t no debe verse afectado por la ocurrencia de un evento en el momento u, cuando $t > u$. Sin embargo, el tipo de covariables dependientes del tiempo que tenemos en los estudios longitudinales no cumplen con esta condición. Esto se debe a que son el resultado de un proceso estocástico generado por el sujeto, el cual está directamente relacionado con el mecanismo que controla que se produzca el evento de interés. En otras palabras, la variable longitudinal no es independiente del evento de interés. Podemos imaginar varias situaciones donde esto ocurre. Supongamos que estamos interesados en estudiar el tiempo hasta que se produzca in evento cardiovascular. Obviamente, si recogemos la variable tensión arterial de forma longitudinal, ambos procesos estarán relacionados y por lo tanto, las condiciones para aplicar el modelo de Cox no se cumplen. En base a esto, para producir inferencias correctas, necesitamos aplicar un modelo conjunto que tenga en cuenta la distribución conjunta de los resultados longitudinales y de supervivencia.

Otra ventaja de estos modelos es que permiten tratar las medidas de error en las variables dependientes del tiempo (variable longitudinal en este caso). En un modelo de Cox con covariables dependientes del tiempo, asumimos que las variables se miden sin error.

> IMPORTANTE: Cuando pensamos en covariables dependientes del tiempo, primero debemos distinguir entre dos categorías diferentes, a saber, covariables internas o endógenas o covariables externas o exógenas. Las covariables internas se generan a partir del propio individuo y por tanto requieren la existencia de dicho individuo. Por ejemplo el recuento de células CD4 y el riesgo de muerte por VIH son procesos estocásticos generados por el individuo. Por otro lado, la contaminación del aire es una covariable externa a los ataques de asma, ya que el paciente no influye en la contaminación del aire.

De esta forma, nos enfrentamos ante dos situaciones en las que queremos usar estos modelos. Primero, cuando nos centramos en el resultado de supervivencia y deseamos tener en cuenta el efecto de una covariable dependiente del tiempo endógena medida con error, y segundo, cuando nos interesamos en que la vari-

able resultado sea la variable longitudinal y deseamos corregir por el abandono no aleatorio (no aleatorio porque no seguimos observando al individuo ya que se ha producido el evento de interés).

8.2 Joint models

Como mencionamos, los 'joint models' tienen en cuenta dos resultados, la respuesta longitudinal y el tiempo de supervivencia. Para estimar este tipo de modelos, primero necesitamos ajustar un modelo para la respuesta longitudinal (generalmente un modelo lineal mixto) y luego para el tiempo de supervivencia. Estos modelos ya los hemos visto en capítulos anteriores, así que ahora lo que haremos es explicar cómo estimar los 'joint models' con R.

Para ilustrar cómo llevar a cabo estos análisis continuaremos con los datos de Sida analizados en el capítulo anterior. Primero necesitamos ajustar por separado el modelo lineal mixto (datos longitudinales) y el modelo Cox (tiempo hasta evento), y luego tomar los objetos devueltos y usarlos como argumentos principales en la función `jointModel()` de la librería `JM`.

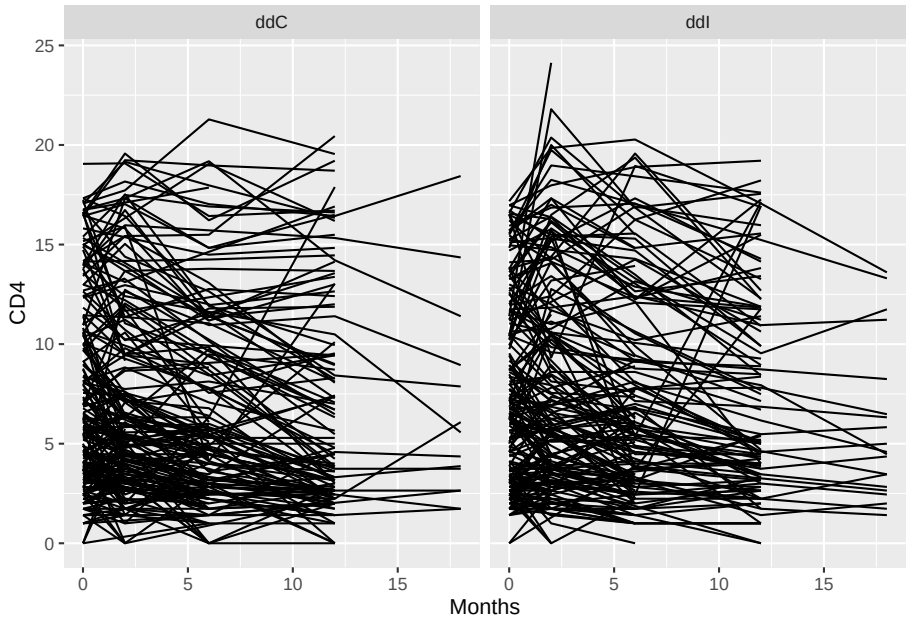
```
library(JM)
data(aids)
head(aids)
```

```
  patient  Time death       CD4 obstime drug
1       1 16.97     0 10.677078       0  ddC
2       1 16.97     0  8.426150       6  ddC
3       1 16.97     0  9.433981      12  ddC
4       2 19.00     0  6.324555       0  ddI
5       2 19.00     0  8.124038       6  ddI
6       2 19.00     0  4.582576      12  ddI
  gender prevOI         AZT start  stop event
1   male   AIDS intolerance     0  6.00     0
2   male   AIDS intolerance     6 12.00     0
3   male   AIDS intolerance    12 16.97     0
4   male noAIDS intolerance     0  6.00     0
5   male noAIDS intolerance     6 12.00     0
6   male noAIDS intolerance    12 18.00     0
```

La idea aquí es probar el efecto del tratamiento sobre la supervivencia después de ajustar el recuento de células CD4, que es una medida recogida a lo largo del tiempo. Este también es el modelo que ajustamos en la sección anterior usando un modelo de Cox con datos dependientes del tiempo, pero que como hemos comentado, no cumplen las condiciones necesarias para que la estimación mediante verosimilitud parcial sea correcta.

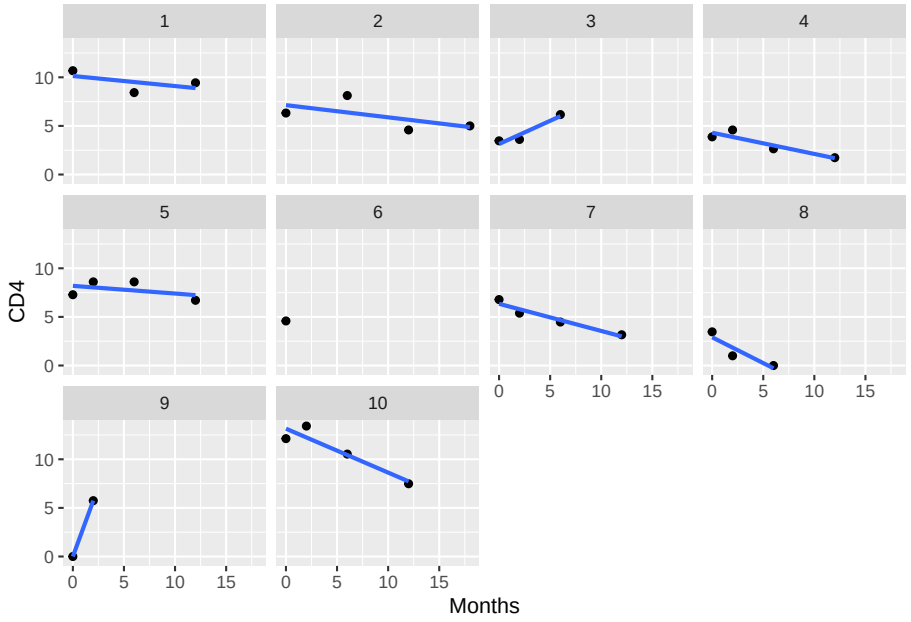
Veamos que valores toma la variable CD4 a lo largo del tiempo usando las funciones de ggplot que ya hemos visto en este curso

```
ggplot(aids, aes(x = obstime, y = CD4, group = patient)) +
  geom_line() + xlab("Months") + facet_wrap(~drug)
```



Visualicemos los 10 primeros individuos para ver si tenemos que usar un modelo mixto con intercept o pendiente aleatoria

```
aids10 <- filter(aids, patient%in%c(1:10))
ggplot(aids10, aes(x = obstime, y = CD4)) +
  geom_point() +  stat_smooth(method = "lm", se = FALSE) +
  xlab("Months") + facet_wrap(~patient)
```



Ahora vamos a especificar y ajustar los modelos para cada uno de nuestros outcomes. El modelo lineal de efectos mixtos para los recuentos de células CD4 incluye:

- Parte de efectos fijos: efecto principal del tiempo y la interacción con el tratamiento.
- Matriz de diseño de efectos aleatorios: el *intercept* y un término de tiempo, ya que vemos en la gráfica anterior que ambos son aleatorios, es decir hay *intercepts* y pendientes distintas para cada individuo.

El submodelo de supervivencia incluye: efecto del tratamiento (como una covariable independiente del tiempo) y el verdadero efecto subyacente del recuento de células CD4 estimado a partir del modelo longitudinal (como dependiente del tiempo). Para el modelo de Cox, asumiremos que la función de riesgo basal es constante por partes (dependiendo de cuando se ha observado los datos longitudinales). Es por ello que definimos `method = "piecesiwse-PH-GH"`. Otras posibilidades incluye estimarla mediante un modelo paramétrico de Weibull ("weibull-AFT-GH") o utilizar B-splines que nos daría una estimación suave de la función de riesgo basal ("spline-PH-GH"). Para más detalles ejectuar `?JointModel`

```
fitLME <- lme(CD4 ~ obstime:drug, random = ~ obstime | patient, data = aids)
fitSURV <- coxph(Surv(Time, death) ~ drug + gender, data = aids.id, x = TRUE)
fitJM <- jointModel(fitLME, fitSURV, timeVar = "obstime", method = "piecewise-PH-GH")
summary(fitJM)
```

```
Call:
```

```
jointModel(lmeObject = fitLME, survObject = fitSURV, timeVar = "obstime",
    method = "piecewise-PH-GH")

Data Descriptives:
Longitudinal Process        Event Process
Number of Observations: 1405    Number of Events: 188 (40.3%)
Number of Groups: 467

Joint Model Summary:
Longitudinal Process: Linear mixed-effects model
Event Process: Relative risk model with piecewise-constant
        baseline risk function
Parameterization: Time-dependent

   log.Lik      AIC      BIC
 -4340.062 8714.123 8784.611

Variance Components:
             StdDev    Corr
(Intercept)  4.5280  (Intr)
obstime      0.1700 -0.0503
Residual     1.8747

Coefficients:
Longitudinal Process
                  Value Std.Err z-value p-value
(Intercept)      7.2059  0.1349 53.4349 <0.0001
obstime:drugddC -0.1897  0.0211 -8.9736 <0.0001
obstime:drugddI -0.1711  0.0217 -7.8684 <0.0001

Event Process
             Value Std.Err z-value p-value
drugddI     0.3548  0.1581  2.2441  0.0248
gendermale -0.2893  0.2606 -1.1101  0.2669
Assoct     -0.3001  0.0382 -7.8480 <0.0001
log(xi.1)  -2.2292  0.3037 -7.3400
log(xi.2)  -1.9447  0.2994 -6.4961
log(xi.3)  -1.6380  0.3384 -4.8407
log(xi.4)  -2.1784  0.4178 -5.2145
log(xi.5)  -2.1003  0.3968 -5.2931
log(xi.6)  -2.0977  0.4691 -4.4715
log(xi.7)  -2.0881  0.5873 -3.5556

Integration:
method: Gauss-Hermite
quadrature points: 15
```

```
Optimization:
Convergence: 0
```

> IMPORTANTE: Debido al hecho de que la función `jointModel` extrae toda la información requerida de estos dos objetos (por ejemplo, vectores de respuesta, matrices de diseño, etc.), en la llamada a la función `coxph()` necesitamos especificar el argumento `x = TRUE`. Con esto, la matriz de diseño del modelo de Cox se incluye en el objeto devuelto.
>
> Además, el argumento principal `timeVar` de la función `jointModel()` se usa para especificar el nombre de la variable dependiente del tiempo en el modelo lineal mixto, que se requiere para el cálculo de este submodelo.

Antes de continuar descriendo los resultados obtenidos, notemos que los resultados son similares al modelo de Cox utilizado en la sección anterior. Es decir, los individuos que toman didanosine (ddI) tienen peor supervivencia que los que toman zalcitabine (ddC) (p=0.0248). Notemos de nuevo como el valor es más significativo que con el modelo de Cox, cumpliéndose así la premisa que analizar los datos con el modelo más adecuado va en favor del investigador.

El parámetro etiquetado 'Assoct' mide el efecto de la variable CD4 (modelada mediante el modelo mixto) en el riesgo de muerte, que en este caso es muy significativa (p<0.0001) y nos indica que este riesgo decrece a medida que aumentan los valores de CD4. Los parametros (xi.1, xi.2, ..) corresponden a los parámetros de la functión de riesgo basal estimada mediante una functión constante por partes

Para obtener el Hazard Ratio tanto de las variables fijas como de las variables longitudinales tenemos que exponenciar el valor que observamos en la tabla. En consecuencia, como este valor para la variable longitudinal CD4 es -0.30, entonces, un aumento de una unidad en el recuento de células CD4 disminuye el riesgo en un 26% (exp(-0.30) = 0.74).

También podemos calcular el IC95% mediante

```
confint(fitJM, parm = "Event")
```

```
                2.5 %       est.     97.5 %
drugddI     0.04492354  0.3547566  0.6645897
gendermale -0.80012212 -0.2893215  0.2214791
Assoct     -0.37504904 -0.3001018 -0.2251546
```

```
exp(confint(fitJM, parm = "Event"))
```

```
               2.5 %      est.    97.5 %
drugddI    1.0459479 1.4258336 1.9436929
gendermale 0.4492741 0.7487714 1.2479212
```

```
Assoct      0.6872556 0.7407428 0.7983928
```

Si queremos ver los efectos para el modelo longitudinal

```
confint(fitJM, parm = "Longitudinal")
```

```
                     2.5 %       est.     97.5 %
(Intercept)      6.9416260  7.2059361  7.4702462
obstime:drugddC -0.2311402 -0.1897056 -0.1482710
obstime:drugddI -0.2137151 -0.1710964 -0.1284777
```

Podemos visualizar las predicciones para un individuo (por ejemplo el 2) tras las 3 o 4 primeras observaciones de CD4 mediante

```
aids.id1 <- filter(aids, patient==2)
fit3 <- survfitJM(fitJM, newdata = aids.id1[1:3, ], idVar = "patient")
fit4 <- survfitJM(fitJM, newdata = aids.id1[1:4, ], idVar = "patient")
par(mfrow=c(1,2))
p1 <- plot(fit3, estimator="mean", include.y = TRUE, conf.int=0.95,
          fill.area=TRUE, col.area="lightblue", main="Patient 2")
p2 <- plot(fit4, estimator="mean", include.y = TRUE, conf.int=0.95,
           fill.area=TRUE, col.area="lightblue", main="Patient 2")
```