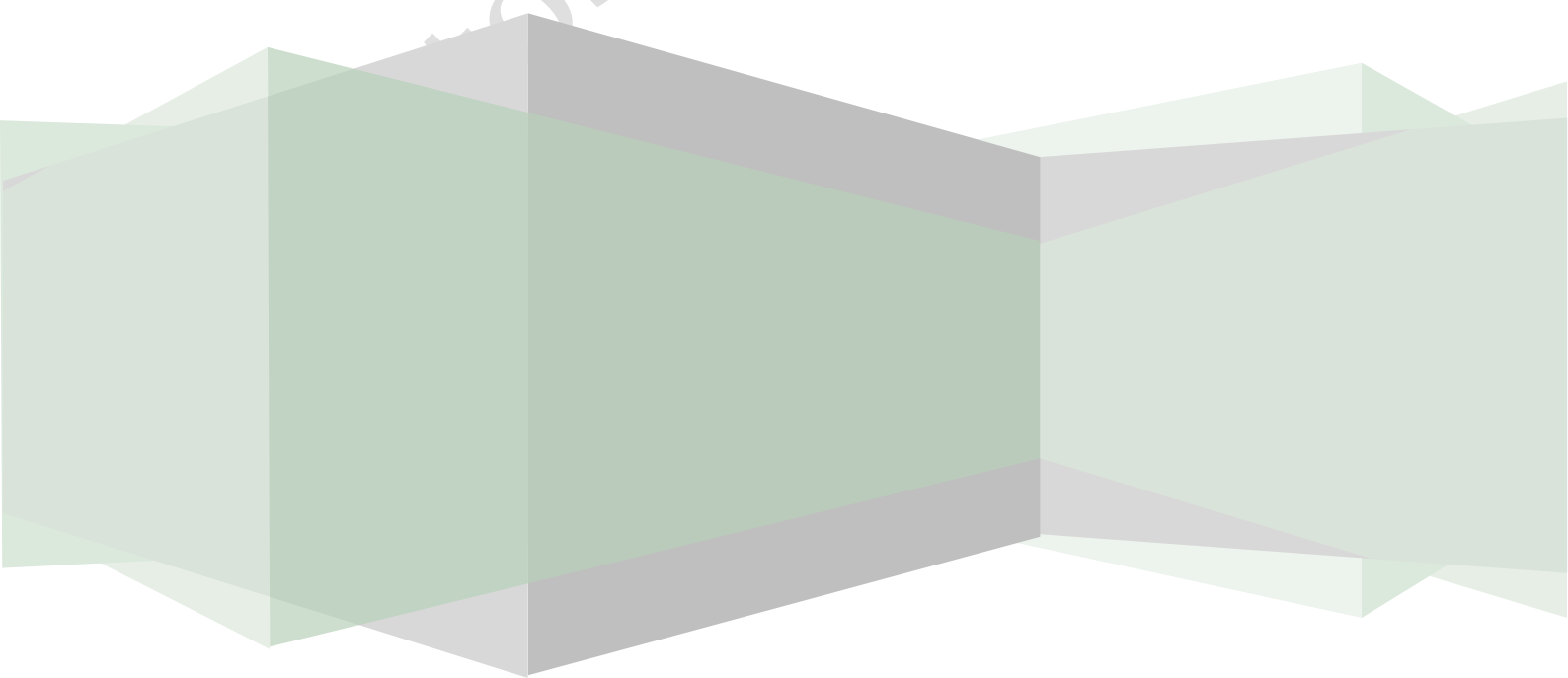


# Datos jerárquicos en Ciencias Ambientales

Ejemplos prácticos y análisis de modelos  
jerárquicos en lenguaje R



Garibaldi, Lucas Alejandro

Datos jerárquicos en ciencias ambientales : ejemplos prácticos y análisis de modelos jerárquicos en lenguaje R / Lucas Alejandro Garibaldi ; Cecilia Casas ; Fernando Biganzoli. - 1a ed. - Bariloche : el autor, 2014.

E-Book.

ISBN 978-987-33-6434-1

1. Estadística. 2. Ciencias Ambientales. I. Casas, Cecilia II. Biganzoli, Fernando III.

Título

CDD 310.4

Cita de la obra

Garibaldi L.A., Casas C., Biganzoli F. 2014. Datos jerárquicos en Ciencias Ambientales: ejemplos prácticos y análisis de modelos jerárquicos en lenguaje R. 1a ed. - Bariloche: el autor. pp.242. ISBN 978-987-33-6434-1

Datos jerárquicos

## Datos jerárquicos en Ciencias Ambientales

Ejemplos prácticos y análisis de modelos jerárquicos en lenguaje R

---

Lucas A. Garibaldi\*, Cecilia Casas\*, Fernando Biganzoli



### Lucas A. Garibaldi

---

Doctor en Ciencias Agropecuarias - UBA  
Profesor regular - UNRN  
Investigador - CONICET  
lgaribaldi@unrn.edu.ar



### Cecilia Casas

---

Doctora en Ciencias Agropecuarias - UBA  
Departamento de Recursos Naturales y Ambiente  
Facultad de Agronomía, UBA  
ccasas@agro.uba.ar



### Fernando Biganzoli

---

Doctor en Ciencias Agropecuarias - UBA  
Departamento de Métodos Cuantitativos y Sistemas de Información  
Facultad de Agronomía, UBA  
biganzol@agro.uba.ar

\*Realizaron una contribución equivalente

Diseño y diagramación: Cecilia Casas

## Dedicatoria

A nuestra querida Susana Perelman, por su apoyo incondicional, por sus consejos desinteresados, por sus enseñanzas, por ser una guía constante y un ejemplo que nos ayuda a crecer y seguir adelante. Todo nuestro afecto y agradecimiento. ¡Muchas gracias!



**Susana Perelman**

---

Magister Scientiae área Biometría; Escuela de Postgrado Facultad de Agronomía. Universidad de Buenos Aires.  
Profesora Asociada (DE). Coordinadora del Departamento de Métodos Cuantitativos Aplicados. FA-UBA.

## Prefacio

Los datos que utilizan los profesionales de las ciencias ambientales y sociales para tomar decisiones y contrastar hipótesis usualmente cuentan con estructura jerárquica (multi-nivel), como mediciones en plantas agrupadas en distintos sitios a su vez agrupados por características climáticas, o el análisis de datos provenientes de individuos agrupados en estudios los cuales pertenecen a distintas regiones. Estos datos no cumplen, al menos, con el supuesto de observaciones independientes para los modelos y análisis estadísticos comúnmente aprendidos en los cursos de grado, entre ellos regresión lineal simple o múltiple y experimentos factoriales (o en bloques) de efectos fijos. Con el objetivo de brindar algunas herramientas que ayuden a superar estas dificultades el Dr. Lucas A. Garibaldi diseñó el curso de posgrado "Datos jerárquicos en ciencias ambientales: colección y análisis con R".

En esta obra recopilamos una serie de prácticas que utilizamos a lo largo del curso. Un aspecto interesante de las prácticas que presentamos es que están basadas en datos reales los cuales llegaron a nosotros a través de amigos y conocidos en búsqueda de un análisis adecuado a sus diseños experimentales. A todos ellos, les agradecemos enormemente la gentileza de brindar sus datos a los fines didácticos del curso y posteriormente, de esta obra. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.

Los capítulos y las prácticas fueron desarrolladas y pensadas con la intención de proporcionar a estudiantes e investigadores de otras disciplinas que no sean la Biometría, los conocimientos básicos sobre cómo coleccionar (diseño de estudios), modelar y analizar datos con estructura jerárquica utilizando el programa R y, limitándonos al estudio de modelos con una sola variable de respuesta (dependiente). Con este objetivo, recomendamos seguir los capítulos frente a una computadora analizando las sentencias y los datos reales que se encuentran disponibles en el sitio <https://sites.google.com/site/datosjerarquicos> solapa

prácticas y que, acompañan la obra. De manera adicional, encontrarán en el anexo las indicaciones necesarias para instalar los programas y herramientas de análisis necesarias.

A lo largo de los capítulos presentamos ejemplos con estructuras jerárquicas de distinta complejidad como diseño en parcelas divididas, diseños anidados, medidas repetidas en el tiempo (autocorrelación) y en el espacio. Utilizamos factores aleatorios, funciones de la varianza y estructuras de correlación para representar adecuadamente los modelos. Abordamos ejemplos distintas distribuciones estocásticas como la normal, la binomial o la poisson a través de modelos mixtos generalizados. Evaluamos la validez de los modelos, la independencia, la bondad de ajuste. Presentamos distintos marcos de inferencia como el frecuentista, evaluación de relaciones de verosimilitud e inferencias multimodelo. Los capítulos no desarrollan los conceptos teóricos. Para ello, sugerimos la lectura de diversos textos publicados que abordan los conceptos presentes en esta obra de manera exhaustiva (Bolker, 2008; Logan M., 2010; Gelman A. y Hill J., 2007, Pinheiro y Bates 2000, Zuur *et al.* 2009). Esta obra tampoco pretende revisar de manera exhaustiva los múltiples modelos de dependencia espacial o temporal de los datos, para lo cual recomendamos realizar cursos o abordar textos específicos (por ejemplo, de geoestadística).

Esperamos que la obra les resulte útil y de ayuda para abordar sus propios análisis.

Lucas A. Garibaldi

Cecilia Casas

Fernando Biganzoli

# Contenido

<b>Dedicatoria</b> .....	<b>4</b>
<b>Prefacio</b> .....	<b>5</b>
<b>Contenido</b> .....	<b>7</b>
<b># CAPÍTULO 0: Funciones básicas</b> .....	<b>9</b>
<b>#CAPÍTULO 1: Edición y manejo de tabla de datos</b> .....	<b>15</b>
<b># CAPÍTULO 2: Explorar los datos de manera gráfica</b> .....	<b>21</b>
<b># CAPÍTULO 3: Modelos y jerarquías</b> .....	<b>33</b>
<b># CAPÍTULO 4: Estimación y selección de modelos</b> .....	<b>43</b>
<b># CAPÍTULO 5: Modelar varianzas</b> .....	<b>63</b>
<b># CAPÍTULO 6: Modelar las varianzas II</b> .....	<b>80</b>
<b># CAPÍTULO 7: Autocorrelación - Medidas repetidas con intervalos iguales</b> .....	<b>109</b>
<b># CAPÍTULO 8: Autocorrelación – Medidas repetidas con intervalos iguales</b> <b>II</b> .....	<b>121</b>
<b># CAPÍTULO 9: Correlación espacial</b> .....	<b>149</b>
<b># CAPÍTULO 10: Correlación espacial - Intervalos desiguales</b> .....	<b>160</b>
<b># CAPÍTULO 11: Modelos con jerarquías en experimentos observacionales</b> .....	<b>171</b>
<b># CAPÍTULO 12: Modelos lineales generalizados mixtos - Binomial</b> .....	<b>194</b>
<b># CAPÍTULO 13: Modelos lineales generalizados mixtos - Conteos</b> .....	<b>210</b>
<b># CAPÍTULO 14: Ejercicio</b> .....	<b>227</b>
<b>Índice de funciones</b> .....	<b>229</b>
<b>Índice de figuras</b> .....	<b>231</b>
<b>Anexo</b> .....	<b>239</b>

# Funciones Básicas

---

## Contenidos

<b># CAPÍTULO 0: Funciones básicas</b> .....	<b>9</b>
<b># 1- Configuración inicial y comandos básicos</b> .....	<b>9</b>
# 1.a- Codificación .....	9
# 1.b- Enviar o NO sentencias a R.....	9
# 1.c- Definir el directorio de trabajo .....	9
# 1.d- Funciones .....	9
# 1.e- Errores .....	10
# 1.f- ¡Ayudas!.....	10
<b># 2- Definir objetos</b> .....	<b>10</b>
<b># 3- Definir vectores</b> .....	<b>11</b>
<b># 4- Definir matrices o tablas de datos</b> .....	<b>11</b>
# 4.a- Importar archivos en formato .csv o .txt.....	11
# 4.b- Leer archivos con funciones específicas.....	12
<b># 5- Funciones básicas: ejemplos</b> .....	<b>12</b>
<b># 6- Resumen</b> .....	<b>13</b>

---

---



## # CAPÍTULO 0: Funciones básicas

### # 1- Configuración inicial y comandos básicos

#### # 1.a- Codificación

# En "Herramientas" > "Opciones" > "General" > "Codificación" = UTF-8  
# Esta configuración evitará problemas de tipografía.  
# Si están usando RStudio, al pie de la ventana que muestra la sentencia encontrarán un índice que muestra los título y subtítulo con jerarquías

#### # 1.b- Enviar o NO sentencias a R

# En R Studio la ventana que leemos es un editor de sentencias.  
# Para ser enviadas a R (la ventana que tenemos debajo), con el cursor sobre la línea a ejecutar, presionamos Ctrl+Enter o el botón "Run" (arriba). Ejemplo:

```
"Hola"  
[1] "Hola"
```

# Todo texto que tenga un signo numeral delante "#" no interfiere con la sentencia.  
# Es decir, R lo ignora.  
# Esto es de utilidad dado que podemos hacer comentarios como los que estamos leyendo en este momento.  
# Y ustedes, podrán agregar sus propios comentarios y apuntes durante el curso.

#### # 1.c- Definir el directorio de trabajo

# Este paso sirve para simplificar la sintaxis posterior.  
# La función "getwd()" nos muestra el directorio desde donde esta leyendo R  
getwd()  
# La función "setwd(...)" define un directorio nuevo. Definimos el directorio donde queremos trabajar  
# setwd("D:\\Mis documentos")

# En RStudio, podemos hacer lo mismo desde la solapa "Files" (Abajo a la derecha). Ubicamos el directorio en la ventana. Luego "More" > "Set as Working Directory"  
# Observar que R ejecutó la sentencia anterior

# Si NO definimos el directorio, debemos escribir la ruta completa en cada sentencia que la necesite.  
# ATENCIÓN: en windows, Podemos poner dos barras invertidas("D:\\Mis documentos\\1-Gusanos\_seda.txt")  
# o, una derecha ("D:/Mis documentos/1\_Gusanos\_seda.txt")

#### # 1.d- Funciones

# R tiene funciones ya definidas (también permite crear nuevas).  
# Las funciones se encuentran agrupadas en paquetes. Al abrir el programa se carga por defecto un paquete "básico".  
# Más adelante en el curso vamos a cargar y usar paquetes específicos.  
# Cada uno de estos paquetes tiene funciones definidas por sus autores.

**# 1.e- Errores**

# Las sentencias aparecen en azul en la ventana de R.  
 # Si la sentencia es correcta, nos dará el resultado y, un signo ">" el final  
 # Si hay un error en la sentencia, aparecerá una leyenda en rojo indicando el problema.

**# 1.f- ¡Ayudas!**

# **¡IMPORTANTE!** La tecla "tab" en RStudio nos ayuda con la sintaxis y especificaciones de las funciones.  
 # Las ayudas en R tienen varias ventajas. Una de ellas, es que se trata de documentos con una estructura estandarizada.  
 # Esto hace que sea rápido encontrar la información que buscamos (ej: ¿Para qué sirve la función?,  
 # ¿En qué paquete se encuentra?, ¿Cuáles son los parámetros que admite y cómo es la sintaxis?)

**# 2- Definir objetos**

```
x<-2 # Definimos el objeto "x"
```

```
x # "x" es un objeto al que le asignamos (<-) el valor "2"
```

```
[1] 2
```

```
y<-"hola" # Usamos comillas para marcar texto
```

```
y
```

```
[1] "hola"
```

```
y<-2013 # Sobrescribimos "y". Ahora contiene el elemento "2013"
```

```
y
```

```
[1] 2013
```

```
# También podemos definir un objeto que contiene operaciones
```

```
z<-3*(5-1)^2
```

```
z
```

```
[1] 48
```

# Un objeto ("x", "y", "z", etc) puede contener un valor o, como veremos en breve, un vector, una tabla de datos, una matriz, un array.

# **¡IMPORTANTE!** R es sensible a las mayúsculas y las minúsculas.

```
Y<-3
```

```
Y
```

```
[1] 3
```

```
y # Lo definimos antes
```

```
[1] 2013
```

```
# Podemos hacer operaciones con objetos
```

```
x*z
```

```
[1] 96
```

# (aunque en este caso los objetos contienen un valor, estas operaciones también son válidas cuando contienen un vector o una matriz)

```
c<-x+z
```

```
c
```

```
[1] 50
```

```
s<-x/3+z
```

```
s
```

```
[1] 48.66667
```

```
s+6
```

```
[1] 54.66667
```

```
c*s
```

```
[1] 2433.333
```

```
e <- (s + 2 * sqrt(s))/(s + 5 * sqrt(s))
```

```
e
```

```
[1] 0.7495021
```

### # 3- Definir vectores

# Un vector es una secuencia de elementos. Tiene una dimensión

```
v<-c(3,5,6,7,32)
```

```
v
```

```
[1] 3 5 6 7 32
```

```
v1<-c(3, "hola", 5)
```

```
v1
```

```
[1] "3" "hola" "5"
```

# Para definir los elementos del vector usamos la función "c" ("concatenar", combina elementos) seguido de paréntesis y los elementos separados entre comas.

### # 4- Definir matrices o tablas de datos

# Las matrices y tablas poseen dos dimensiones (filas y columnas)

# A diferencia de las matrices, las tablas de datos poseen tanto valores numéricos como categóricos.

# Las matrices y las tablas pueden ser generadas en R.

# Pero, R también puede leer datos importados desde otros formatos (ej.: .txt, .csv, etc).

# Vamos a trabajar importando la tabla que contiene los datos que queremos analizar.

# Existen varias formas de leer tablas de datos. Veamos algunas.

#### # 4.a- Importar archivos en formato .csv o .txt

# Desde la solapa "Environment" (arriba a la derecha) > "Import Dataset" > "From Text File..."

# De esta manera importamos los datos a un objeto con el nombre que indicamos en la ventana para importar.

# Si observamos las sentencias que aparecen en la ventana de R al hacer esto, leemos los comandos que ejecuta RStudio,

# y podemos reproducirlas con el archivo "Gusanos\_seda":

```
# > Gusanos_seda <- read.csv("C:/Documents and
Settings/Cecilia/Escritorio/1_Gusanos_seda.txt")
# > View(Gusanos_seda)
```

#### # 4.b- Leer archivos con funciones específicas

# Funciones "read.csv" y "read.table"

# Si el archivo que deseamos leer se encuentra en el directorio que establecimos al comenzar la práctica, la sentencia se simplifica.

# De lo contrario, debemos escribir el directorio completo con la ubicación del archivo.

# Leemos un archivo con formato .csv

```
gusan<-read.csv("1_Gusanos_seda.csv", header=T, sep=",")
```

# "gusan" es el nombre que asignamos al objeto que va a contener el archivo.

# Las funciones como "read.csv" admiten argumentos que separamos por comas.

# En este caso, los argumentos son: el archivo, el encabezado (header= TRUE o, FALSE) y el símbolo que separa las columnas (sep="...").

# En este caso, la separación entre columnas es "," (también pueden ser ";" o "/" para tabulación).

# Vemos la tabla de datos con la función View:

```
View(Gusan)
```

# También podemos leer un archivo con formato .txt.

```
gusan2<-read.table("1_Gusanos_seda.txt", header=TRUE, sep=",")
```

```
View(gusan2)
```

# Función "read.delim"

# Otra alternativa es utilizar la función "read.delim".

# Para ello debemos abrir la tabla de datos en una planilla de cálculos ("LibreOffice", "Excel", etc.),

# seleccionamos las celdas con las que deseamos trabajar, copiamos y ejecutamos la siguiente sentencia:

```
gusan3<-read.delim("clipboard",dec=".", header=TRUE)
```

```
View(gusan3)
```

#### # 5- Funciones básicas: ejemplos

# Aplicamos algunas de las funciones que se encuentran en el paquete básico de R

# Para el vector "v"

```
mean(v)          # la media aritmética
[1] 10.6
```

```
median(v)        # la mediana
[1] 6
```

```
sd(v)            # el desvío estándar
[1] 12.05404
```

```
length(v)        # la longitud o cantidad de elementos que contiene el objeto
[1] 5
```

```
# Para el archivo "1_Gusanos_seda"  
head(gusan) # muestra las primeras seis filas de gusan  
  
names(gusan) # muestra los nombres de las columnas  
  
dim(gusan) # muestra las dimensiones (cantidad de filas y columnas) de la  
tabla de datos gusan  
  
str(gusan) # muestra la estructura: para cada columna (variable), el tipo de  
dato de contiene y la descripción.  
  
summary(gusan) # muestra un resumen de la información contenida en la tabla.  
# Estas funciones sirven para identificar la presencia de NA's (datos no  
disponibles: non available) y el contenido y tipo de dato de cada variable  
(columna).  
# En los próximos capítulos, veremos cómo editar y manejar las tablas de datos.
```

### # 6- Resumen

```
# En esta práctica hemos aplicado una selección de funciones básicas en R que  
utilizaremos durante las siguientes prácticas.  
# Hemos utilizado algunos códigos básicos de configuración, utilizado la ayuda y  
creado objetos.  
# También utilizamos funciones básicas sobre estos objetos así como,  
importamos planillas que por ejemplo pueden contener los datos que deseamos  
analizar.  
# En la próxima práctica aplicaremos funciones de edición y manejo de tablas  
como por ejemplo, seleccionar un subconjunto de datos.
```

# Edición y manejo de tablas de datos

---

## Contenidos

<b>#CAPÍTULO 1: Edición y manejo de tabla de datos</b> .....	<b>15</b>
<b># 1- Configuración inicial</b> .....	<b>15</b>
# 1.a- Definir el directorio de trabajo .....	15
# 1.b- Leer un archivo de datos.....	15
<b># 2- Explorar la tabla de datos</b> .....	<b>15</b>
<b># 3- Indexar</b> .....	<b>16</b>
# 3.a- Operador básico para tablas de datos.....	16
# 3.b- Operador general .....	16
# 3.c- Función "subset" .....	17
<b># 4- Buscar valores</b> .....	<b>17</b>
<b># 5- Guardar archivos</b> .....	<b>18</b>
<b># 6- Especificar el tipo de dato</b> .....	<b>18</b>
<b># 7- NAs: datos no disponibles</b> .....	<b>18</b>
<b># 8- Resumen</b> .....	<b>19</b>

---

---

## #CAPÍTULO 1: Edición y manejo de tabla de datos

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer un archivo de datos

Leemos el archivo "1\_Gusanos\_seda"

```
gusan<-read.csv("1_Gusanos_seda.csv", header=T, sep=",")
```

### # 2- Explorar la tabla de datos

```
str(gusan) # muestra la estructura del objeto "gusan"
```

```
'data.frame': 180 obs. of 10 variables:
 $ n_caja      : int  1 1 1 1 1 1 1 1 1 1 ...
 $ caja       : Factor w/ 6 levels "A","B","C","D",...: 1 1 1 1 1 1 1 1 1 1
 .
 $ linea      : Factor w/ 2 levels "EC","Eoro": 1 1 1 1 1 1 1 1 1 1
 $ n_cap      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ genero     : Factor w/ 5 levels "H","H_MUERTA",...: 1 1 1 3 1 3 3 1 1 3
 .
 $ peso_pupa  : num  0.8 0.93 0.94 0.84 0.99 0.69 0.77 1.11 0.91 0.87
 $ peso_larva : num  2.51 2.2 2.91 2.42 2.2 2.56 3 2.49 2.82 2.9 .
 $ peso_cap   : num  1.04 1.15 1.12 1.06 1.19 0.88 1.01 1.35 1.11 1.11
 $ peso_corteza: num  0.24 0.22 0.18 0.22 0.2 0.19 0.24 0.24 0.2 0.24
 $ seda_bruta : num  23.1 19.1 16.1 20.8 16.8 ...
```

```
summary(gusan) # muestra un resumen con el tipo de información que contiene "gusan"
```

# Identificamos el tipo de variables que contiene y verificamos la correcta lectura de la tabla de datos.

# Estos comandos nos sirven para identificar la presencia de NA's (datos no disponibles: non available) y el contenido y tipo de dato de cada variable (columna).

# Las variables pueden ser numéricas, enteros o factores.

```
n_caja      caja      linea      n_cap      Genero      peso_pupa
Min.:      1 A: 30 EC : 90 Min. :      1 H :      79 Min. :      0.43
1st Qu.:    2 B: 30 Eoro: 90 1st Qu.:    8 H_MUERTA: 1 1st Qu.:    0.8
Median :    3.5 C: 30      Median : 15.5 M :      93 Median :    0.915
Mean :    3.5 D: 30      Mean : 15.5 muerto : 2 Mean :    0.9114
3rd Qu.:    5 E: 30      3rd Qu.: 23 no_se_vio: 5 3rd Qu.:    1
Max. :    6 F: 30      Max. :    30      Max. :    1.49
                                     NA's :    2
```

```
peso_larva  peso_cap      peso_corteza  seda_bruta
Min. :    0.43 Min. :    0.49 Min. :    0.05 Min. :    9.09
1st Qu.:    2.3 1st Qu.:    1.01 1st Qu.:    0.18 1st Qu.:   15.57
Median :    2.6 Median :    1.11 Median :    0.2 Median :   18.18
Mean :    2.525 Mean :    1.11 Mean :    0.1987 Mean :   18.02
```

```
3rd Qu.: 2.873 3rd Qu.: 1.218 3rd Qu.: 0.23 3rd Qu.: 20.36
Max.      : 3.48 Max.      : 1.69 Max.      : 0.28 Max.      : 26.32
          NA's      : 2 NA's      : 2 NA's      : 2
```

# Puede ser que R no esté interpretando de manera correcta el tipo de variable.  
# ¡Este es el momento para hacer las correcciones!

### # 3- Indexar

#### # 3.a- Operador básico para tablas de datos

# El signo "\$" es un operador básico que ubica columnas dentro de una tabla de datos

```
gusan$linea
```

```
summary(gusan$linea)
```

```
 EC  Eoro
 90   90
```

table(gusan\$linea) # genera una tabla de contingencia con la cantidad de datos para la combinación de factores incluidos en la sentencia.

```
 EC  Eoro
 90   90
```

```
summary(gusan$peso_larva)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.430  2.300   2.600   2.525  2.872   3.480
```

```
mean(gusan$peso_larva)
```

```
[1] 2.524944
```

```
median(gusan$peso_larva)
```

```
[1] 2.6
```

```
sd(gusan$peso_larva)
```

```
[1] 0.5375991
```

```
tapply(gusan$peso_larva, gusan$linea, mean)
```

```
 EC      Eoro
2.708667 2.341222
```

# Esta función aplica una función (último argumento: mean, median, sd, etc.) a la variable dada en el primer argumento agrupada según la variable dada en el segundo argumento.

```
tapply(gusan$peso_larva, gusan$linea, sd)
```

```
 EC      Eoro
0.2899795 0.6549189
```

```
tapply(gusan$peso_larva, gusan$linea, length)
```

```
 EC  Eoro
 90  90
```

#### # 3.b- Operador general

# Los corchetes, "[ ]", se utilizan con tablas de datos y matrices.

# A la izquierda de la coma indican las filas y, a la derecha, las columnas



```
gusan[1:5, ]
```

# Cuando dejamos el espacio vacío antes o después de la coma, significa todas las filas o columnas, respectivamente.

	n_caja	caja	linea	n_cap	genero	peso_pupa	peso_larva	peso_cap	peso_corteza	seda_bruta
1	1	A	EC	1	H	0.8	2.51	1.04	0.24	23.08
2	1	A	EC	2	H	0.93	2.2	1.15	0.22	19.13
3	1	A	EC	3	H	0.94	2.91	1.12	0.18	16.07
4	1	A	EC	4	M	0.84	2.42	1.06	0.22	20.75
5	1	A	EC	5	H	0.99	2.2	1.19	0.2	16.81

```
gusan[3,7]
```

```
[1] 2.91
```

```
gusan[c(3,6:7), c(1,5)]
```

	n_caja	genero
3	1	H
6	1	M
7	1	M

```
gusan[gusan$caja=="A", 2]
```

```
gusan[gusan$caja=="A", "peso_cap"]
```

### # 3.c- Función "subset"

```
subset(gusan, caja=="A"|caja=="C") # | = unión
```

# la función "subset" tiene por argumentos:

# el objeto que contiene la planilla de datos

# y la expresión para seleccionar un subconjunto de datos.

```
subset(gusan, caja=="D"& peso_larva > 1.1) # & = intersección
```

```
summary(subset(gusan, caja=="D" & peso_larva>1.1))
```

```
resu<-subset(gusan, caja=="A" & peso_larva>1.1)
```

```
summary(resu)
```

```
dim(resu)
```

# En el resumen ("summary") detectamos que hay datos que NO nos interesan dentro de la variable "genero".

# Nos quedamos solo con los que nos interesan: "M" y "H"

```
gusa2<-subset(gusan, genero=="M"|genero=="H")
```

```
summary(gusa2)
```

### # 4- Buscar valores

```
max(gusan$peso_larva)
```

```
[1] 3.48
```

```
which.max(gusan$peso_larva)
```

```
[1] 169
```

# ¡ATENCIÓN! La función "which.max" indica la posición: la fila donde se encuentra el valor máximo.

# También podemos aplicar las funciones "min" y "which.min"

```
gusan$caja=="C" # Respuesta lógica: cumple o no con la condición
(TRUE/FALSE)
```

### # 5- Guardar archivos

```
write.csv(gusa2, file="gusanos.csv")
# Guardamos esta tabla como un nuevo archivo
# Se guardará en el directorio que hemos seleccionado anteriormente.
# Si no recordamos cuál es, podemos usar "getwd()"
# Lo buscamos en la carpeta
```

### # 6- Especificar el tipo de dato

```
# Es frecuente utilizar números para identificar categorías.
# Por ejemplo, bloques (1, 2, 3, etc.) que contienen parcelas (1, 2, 3, etc.).
# R suele interpretar estos datos como una variable numérica continua!
# Para transformar en factores (categorías) valores numéricos usamos la función
"as.factor"
```

```
# Esta situación se da por ejemplo con la variable "n_caja"
gusan$n_caja<-as.factor(gusan$n_caja)
```

```
# Dado que indicamos a la izquierda de asignación (<-) el nombre de una
columna que ya existe, la sobrescribimos.
# También podemos utilizar la función "factor(variable)" al escribir los modelos
summary(gusan)
```

n_caja	caja	Linea
01:30	A:30	EC :90
02:30	B:30	Eoro:90
03:30	C:30	
04:30	D:30	
05:30	E:30	
06:30	F:30	

```
# Si queremos hacer la transformación inversa: de factor a numérico, usamos la
función "as.numeric"
```

### # 7- NAs: datos no disponibles

```
# La presencia de NAs en la tabla de datos puede generar errores en algunas
funciones.
# Por ejemplo, que un modelo no pueda ser estimado.
# ¡Usamos el help!
?na.omit
gusan.na<-na.omit(gusan)
# generamos un nuevo objeto "gusan.na" definido por nosotros que contiene el
resultado de la función "na.omit(gusan)"
# La función "na.omit" remueve las filas que contienen NA en al menos una
columna y por lo tanto se reduce el n (longitud) de la serie de datos.
```

```
dim(gusan)
[1] 180 10
```

```
dim(gusan.na)
```

```
[1] 178 10
```

```
# De manera alternativa, algunas funciones gráficas
```

```
# y de modelo permiten incluir el argumento "na.rm=T"
```

### # 8- Resumen

```
# En esta práctica hemos aplicado funciones específicas para indexar tablas,  
seleccionar subconjuntos de datos aplicando distintas funciones así como  
múltiples criterios de selección.
```

```
# Hemos buscado datos específicos y su ubicación en la tabla de datos.
```

Datos jerárquicos

# Explorar los datos de manera gráfica

---

## Contenidos

<b># CAPÍTULO 2: Explorar los datos de manera gráfica</b> .....	<b>21</b>
<b># 1- Configuración inicial</b> .....	<b>21</b>
# 1.a- Definir el directorio de trabajo .....	21
# 1.b- Leer el archivo "gusanos.csv" .....	21
<b># 2- El caso: Producción de seda</b> .....	<b>21</b>
# 2.a- Pregunta de interés .....	22
# 2.b- Variable respuesta.....	22
# 2.c- Variables predictoras.....	22
<b># 3- Gráficos preliminares</b> .....	<b>22</b>
<b># 4- Homogeneidad</b> .....	<b>23</b>
<b># 5- Normalidad</b> .....	<b>24</b>
<b># 6- Presencia de ceros</b> .....	<b>26</b>
<b># 7- Multilinealidad</b> .....	<b>26</b>
# 7.a- Determinante en la matriz de correlación.....	26
# 7.b- Factor de Inflación de la Varianza.....	27
# 7.c- Matriz de correlación gráfica.....	28
# 7.d- Alternativa para reducir la multilinealidad.....	29
<b># 8- Interacciones</b> .....	<b>30</b>
<b># 9- Independencia</b> .....	<b>31</b>
<b># 10- Resumen</b> .....	<b>31</b>
<b># 11- Agradecimiento</b> .....	<b>31</b>

---



---

## # CAPÍTULO 2: Explorar los datos de manera gráfica

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo "gusanos.csv"

# Aclaración: abrir el archivo "gusanos.csv" que generamos en el CAPÍTULO 1 utilizando alguna de las maneras vistas.

```
# gusa<-.....
summary(gusa)
```

### # 2- El caso: Producción de seda

La producción de seda a partir del gusano de seda *Bombyx morio* se inició en China cerca del 2700 a.C. En la actualidad, la producción de seda es un negocio creciente en el mundo siendo China e India los principales productores. En nuestro país la producción de seda tiene aproximadamente 250 años y se inició de la mano del jesuita Ramón María de Termeyer. Después de varios años de idas y vueltas respecto del apoyo político, en el año 2003 fue promulgada la ley 25.747, que prevé la creación de estaciones sericícolas para estudiar la producción: propagar el cultivo de la morera, determinar las mejores variedades del gusano; promocionar la venta e industrialización de los capullos y potenciar el cooperativismo de los productores.



Figura 2. 1: esquema del diseño experimental que evalúa la producción de seda de dos líneas de gusanos de seda.

El mantenimiento de razas puras de gusano de seda es muy importante para encontrar los caracteres deseados para la producción de huevos de híbridos

comerciales (Mubashar et al, 2010). En el marco de un proyecto de mejoramiento genético y aclimatación de líneas genéticas de gusanos de seda, un grupo de investigadores del Laboratorio de sericultura de la Cátedra de Producciones Animales Alternativas de la FAUBA, evaluaron la producción de seda de dos líneas de gusanos ingresadas al país. El objetivo del trabajo fue caracterizar líneas de gusano de seda de origen japonés y chino (E oro y E CH, respectivamente), endocriadas en la FAUBA durante 4 generaciones para evaluar si pueden ser utilizadas en la formación de híbridos comerciales adaptados a la región pampeana. Los gusanos de seda de las líneas E oro y E CH crecieron bajo condiciones controladas de luz, fotoperiodo, humedad y alimentación. Los gusanos fueron colocados en cajas separadas por líneas (30 gusanos por caja en 3 cajas por línea. Fig. 2.1).

### # 2.a- Pregunta de interés

# ¿Qué características de los gusanos podrían predecir la producción de seda?

### # 2.b- Variable respuesta

# Peso de la corteza (estimador de la producción de seda).

# La corteza es la parte externa del capullo y se calcula como "capullo - pupa = corteza"

### # 2.c- Variables predictoras

# - línea: línea genética de los gusanos de seda

# - genero

# - peso larva (gramos)

# - peso capullo (gramos)

# - peso pupa (gramos)

### # 3- Gráficos preliminares

```
with(gusa, boxplot(peso_corteza, xlab="Conjunto de datos", ylab="Peso corteza (g)"))
```

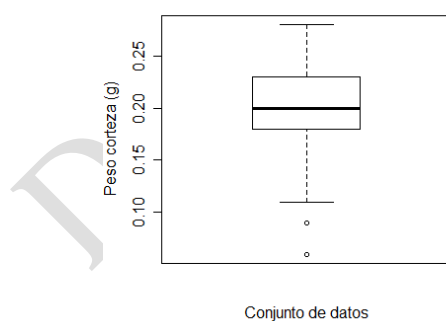


Figura 2. 2: boxplot de la distribución de los valores de peso de la corteza (g/gusano) de los gusanos de seda

# ¡ATENCIÓN! Este gráfico NO indica que los valores por fuera del bigote sean outliers!!

```
with(gusa, dotchart(peso_corteza, xlab="Peso corteza (g)", ylab="Conjunto de datos"))
```

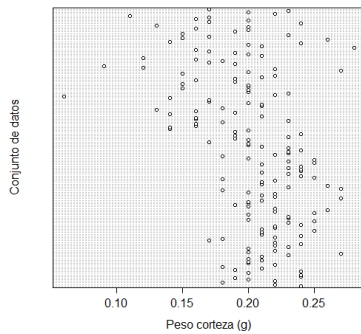


Figura 2. 3: distribución de los valores de peso de la corteza (g/gusano) de los gusanos de seda

```
with(gusa, dotchart(peso_corteza, xlab="Peso corteza (g)", ylab="Conjunto de
datos por línea", groups=línea))
```

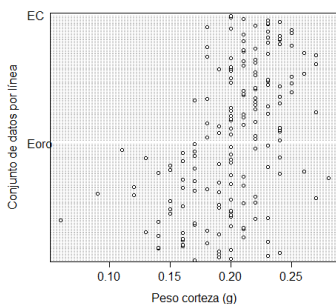


Figura 2. 4: distribución de los valores de peso de la corteza (g/gusano) de los gusanos de seda agrupados por línea genética (EC o Eoro).

# Este gráfico muestra el valor observado (x) en función del número de fila (y) en la tabla de datos.

# Observamos que hay solo dos datos menores de 0.10 y que ambos pertenecen a la línea Eoro.

# Observamos que hay muchos valores alrededor de 0.2 y unos pocos mayores a 0.25 en ambas líneas.

# Este tipo de gráfico permite observar la distribución de los datos y la presencia de valores extremos de manera fácil.

# **ACLARACIÓN:** Esta función se llamaba "dotplot". Luego le cambiaron el nombre a "dotchart".

# **DIAGNÓSTICO:** no tenemos valores extremos o outliers

# Si los gráficos no se ven bien en la ventana de RStudio pueden llevarlos a una ventana nueva usando el botón "Zoom"

#### # 4- Homogeneidad

# **¡ATENCIÓN!** NO lo utilizamos para diagnóstico, solo a modo exploratorio  

```
with(gusa, boxplot(peso_corteza~línea, xlab="Línea genética", ylab="Peso
corteza (g)"))
```

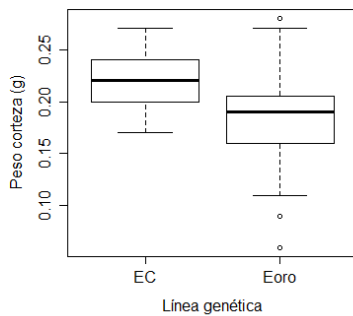


Figura 2. 5: boxplot de los valores de peso de la corteza (g/gusano) de los gusanos de seda agrupados por línea genética (EC o Eoro).

# En este caso, no nos centramos en el valor medio sino que nos interesa que la dispersión de los datos alrededor de la media (el tamaño de las líneas verticales) sea similar entre los grupos.

# Es decir, que sus varianzas sean homogéneas

# Observamos que la dispersión en EC es menor que en Eoro.

# Una conclusión similar obteníamos del dotchart realizado arriba.

# **DIAGNÓSTICO:** Aparentemente las varianzas no serían homogéneas

```
with(gusa, boxplot(peso_corteza~factor(caja), xlab="Caja", ylab="Peso corteza (g)"))
```

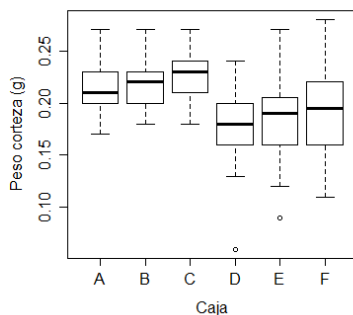


Figura 2. 6: boxplot de los valores de peso de la corteza (g/gusano) de los gusanos de seda agrupados por caja (A, B, C, D, E, o F).

# Un análisis similar al anterior podemos hacer para las cajas.

### # 5- Normalidad

# **¡ATENCIÓN!** NO lo utilizamos para diagnóstico, solo a modo exploratorio

# No todos los análisis asumen distribución normal y otros son bastante robustos a la falta de normalidad

```
library("lattice")
```

# lattice es un paquete que sirve para hacer gráficos.

# La sintaxis tiene algunas diferencias respecto del paquete básico.

```
with(gusa, histogram(~peso_corteza|linea, xlab= "Peso corteza (g)", ylab= "Frecuencia"))
```



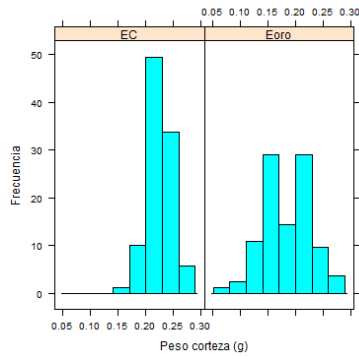


Figura 2.7: histograma de la distribución del peso de la corteza (g/gusano) de los gusanos de seda agrupados por línea genética (EC o Eoro).

```
# Es igual si usamos el argumento "data"
histogram(~peso_corteza|linea, type = "count", data= gusa, xlab= "Peso corteza
(g)", ylab= "Frecuencia")
```

```
# Otros argumentos: type = "count" o "percent"
with(gusa,histogram(~peso_corteza|factor(caja)))
```

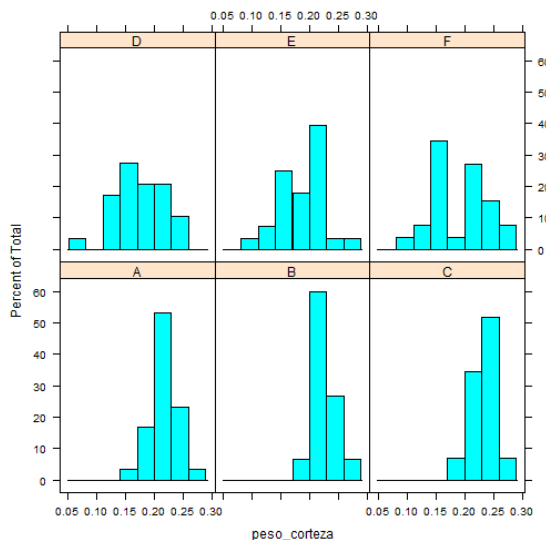


Figura 2. 8: histograma de la distribución del peso de la corteza (g/ gusano) de los gusanos de seda agrupados por línea cajas (A, B, C, D, F).

# **ACLARACIÓN:** siempre analizar los supuestos de homogeneidad en las varianzas y normalidad sobre los residuos del modelo elegido.

# **VENTAJAS:** tenemos más "potencia" al evaluar todos los residuos

# los residuos sirven para diagnosticar el modelo en general, por ejemplo heterogeneidad de varianzas.

# Entonces, los supuestos del modelo los analizaremos sobre los residuos luego de ajustar el modelo.

# Por ahora aprendemos gráficos exploratorios útiles pero esta NO es una guía de evaluación de supuestos de los modelos (lo veremos más adelante).

### # 6- Presencia de ceros

```
plot(table(round(gusa$peso_corteza, 2)), # round= aproximado al MENOR número
con los decimales especificados
type = "h", # type="h", la línea con la altura del valor
xlab = "Observados", ylab = "Frecuencia")
```

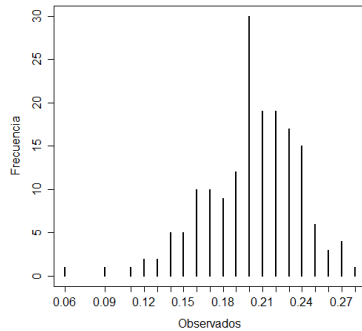


Figura 2. 9: frecuencia de valores del peso de la corteza (g/ gusano) de los gusanos de seda.

```
# Este tipo de gráfico tiene utilidad en casos puntuales.
# Por ejemplo, si trabajamos con conteos.
```

### # 7- Multicolinealidad

```
# Un aspecto importante a evaluar es la multicolinealidad o,
# correlación entre variables predictoras del modelo.
# La multicolinealidad genera un aumento de las varianzas de los estimadores de
los coeficientes de regresión y, en consecuencia, las estimaciones son poco
precisas.
# Métodos de evaluar multicolinealidad
```

#### # 7.a- Determinante en la matriz de correlación

```
# La matriz de correlación tiende a la matriz identidad (su determinante tiende a
la unidad) cuando no hay multicolinealidad fuerte.
# Por el contrario, la matriz de correlación dista de ser la identidad (la
determinante tiende a cero) cuando hay indicios de multicolinealidad.
```

```
# En este caso,
cor.matrix<-cor(gusa[,c("peso_pupa", "peso_larva", "peso_cap")], method=
"pearson", use= "pairwise.complete.obs")
# Observar que solo incluimos predictores numéricos: linea y genero quedaron
excluidos
# El argumento "use= "pairwise.complete.obs"" indica que estime las
correlaciones de las observaciones completas.
# De lo contrario, no podrá estimar las correlaciones y la matriz contendrá NA
```

```
round(cor.matrix, 2) # con la función "round" aproximamos a una cantidad dada
de decimales. Podemos probar con 3 decimales.
```

```
      peso_pupa peso_larva peso_cap
peso_pupa      1.00    -0.18    0.98
peso_larva    -0.18     1.00   -0.11
peso_cap       0.98    -0.11     1.00
```

```
det(cor.matrix) # calcula la determinante de la matriz
[1] 0.03878245
```

# **DIAGNOSTICO:** Existe multicolinealidad en esta matriz de datos  
 # En este caso, que la matriz es pequeña, es fácil de observar la correlación de 0.98 entre peso\_pupa y peso\_cap

### # 7.b- Factor de Inflación de la Varianza

# Es un índice que analiza la colinealidad que produce una determinada variable predictora.

# En la sintaxis de R podemos determinarlo de la siguiente manera:

# Genero un modelo (puede ser lm o, glm) y calculo el VIF con la función vif(modelo) del paquete "car".

```
coli<-lm(peso_corteza ~ peso_pupa+peso_larva+peso_cap, data=gusa)
```

# En e próximo capítulo veremos en detalle cómo se especifican modelos.

```
library("car")
```

```
vif(coli)
```

```
peso_pupa  peso_larva  peso_cap
25.466256  1.135046   24.975278
```

# El mensaje de alerta:

# Warning message:

# In summary.lm(object) : essentially perfect fit: summary may be unreliable

# sugiere que hay problemas en la estimación del modelo. Esto se debe justamente a la colinealidad entre variables predictoras

# "peso\_pupa" con VIF = 25.46 y "peso\_cap" con VIF = 24.95

# En general, se considera que existen problemas de colinealidad si algún FIV es superior a 10, que corresponde a un  $R^2 \sim 0,9$ .

# Sin embargo, existe controversia al respecto (ver Fox 1992, O'Biren 2007, Zuur et al. 2009).

# La colinealidad moderada o alta puede generar conclusiones erróneas sobre todo cuando los efectos son débiles.

# En estos casos, incluso un VIF de 2 puede generar estimaciones no significativas de los parámetros,

# comparado con el modelo sin colinealidad.

# **IMPORTANTE:** La elección de las covariables a excluir se puede basar en los valores de VIF, o tal vez mejor,

# en el sentido común o el conocimiento biológico (Zuur et al. 2010).

# Es esperable encontrar colinealidad cuando se incluyen covariables temporales (por ejemplo, mes, año) o espaciales (por ejemplo, latitud, longitud) con covariables como la temperatura, las precipitaciones, etc.

# En estos casos se deben graficar todas las covariables en contra covariables temporales y espaciales.

# Si avanzamos con el procedimiento, podemos sacar el predictor de mayor VIF y recalculamos el vif del nuevo modelo.

```
which.max(vif(coli))
peso_pupa
1
```

```
coli2<-update(coli, . ~ . - peso_pupa)
vif(coli2)
peso_larva    peso_cap
1.012511      1.012511
```

# Una vez que no hay evidencias de multicolinealidad con el VIF, podemos recalcular la determinante de la nueva matriz de correlación.

# Este procedimiento puede ser útil cuando trabajamos con modelos que contienen muchas variables predictoras posibles.

# **IMPORTANTE:** este no es un procedimiento de SELECCIÓN DE MODELOS o VARIABLES.

# Es clave saber que no es correcto estimar un modelo cuyas variables predictoras tienen gran correlación entre sí. Cuáles variables usar en estos casos responde a un marco conceptual biológico - social y NO HAY UNA RECETA ESTADÍSTICA.

### # 7.c- Matriz de correlación gráfica

# Varios gráficos son útiles para analizar correlaciones. Sólo damos algunos convenientes y útiles.

# En nuestro ejemplo, covariables podrían ser: "peso\_pupa", "peso\_larva", "peso\_cap" ("genero" queda excluida por ser categórica)

```
pairs(gusa[,c("peso_pupa", "peso_larva", "peso_cap")])
```

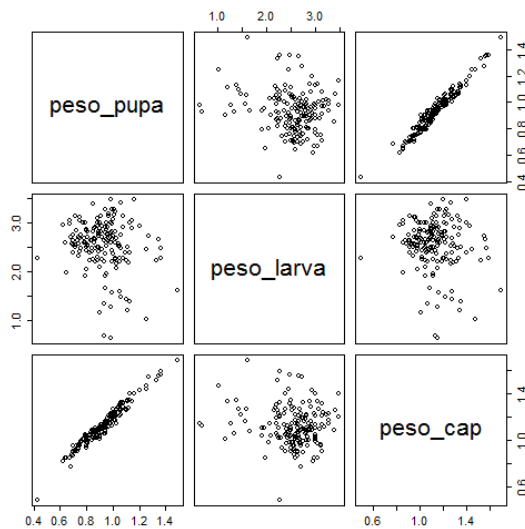


Figura 2. 10: relación entre variables cuantitativas: peso de las pupas, peso de las larvas y peso de los capullos (todas expresadas en g/gusano de seda).

# Podemos agregar los valores de la matriz de correlación al gráfico anterior

```
panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, method="pearson", use="pairwise.complete.obs"))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
```

```

txt <- paste(prefix, txt, sep="")
if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
text(0.5, 0.5, txt, cex = cex.cor * r)
}

# Podemos agregar la línea que muestra la función de regresión entre las
variables.
panel.linea = function(x, y, ...) {
  tmp <- lm(y ~ x, na.action = na.omit)
  abline(tmp)
  points(x, y) }

pairs(gusa[,c("peso_pupa", "peso_larva", "peso_cap")],
      lower.panel = panel.cor,
      upper.panel=panel.linea,
      cex.labels=1,
      labels=c("peso_pupa", "peso_larva", "peso_cap"))

```

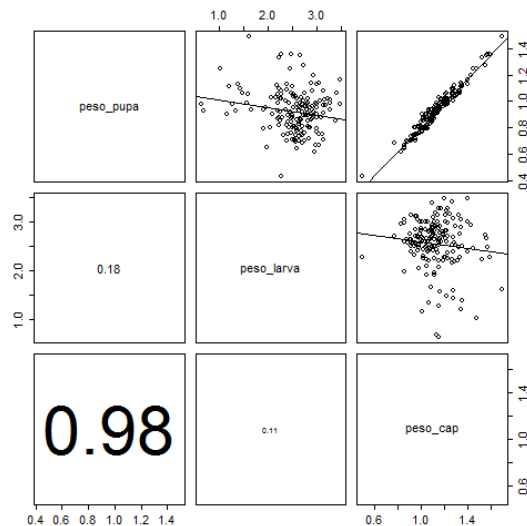


Figura 2. 11: relación entre variable cuantitativas: peso de las pupas, peso de las larvas y peso de los capullos (todas expresadas en g/gusano de seda). Distribución de los puntos en el triángulo superior de la matriz y valor de coeficiente de Pearson en el triángulo inferior de la matriz.

# **DIAGNÓSTICO:** Existe colinealidad (alta correlación) entre peso\_pupa y peso\_capullo.  
# Por lo tanto, no podremos incluir a ambas variables como predictoras en nuestro modelo.  
# Utilizaremos: peso\_cap

#### # 7.d- Alternativa para reducir la multicolinealidad

# Una de las maneras de reducir la multicolinealidad es suprimiendo variables que están correlacionadas con otras.  
# En este caso, la pérdida de capacidad explicativa será pequeña.

## # 8- Interacciones

# El gráfico "coplot" permite observar la presencia de potenciales interacciones considerando tres variables.

```
with(gusa, coplot(peso_corteza ~ peso_larva | factor(caja), show.given=T,
  ylab = "Peso de la corteza(g)",
  xlab = "Peso de la larva (g)",
  panel = panel.linea))
```

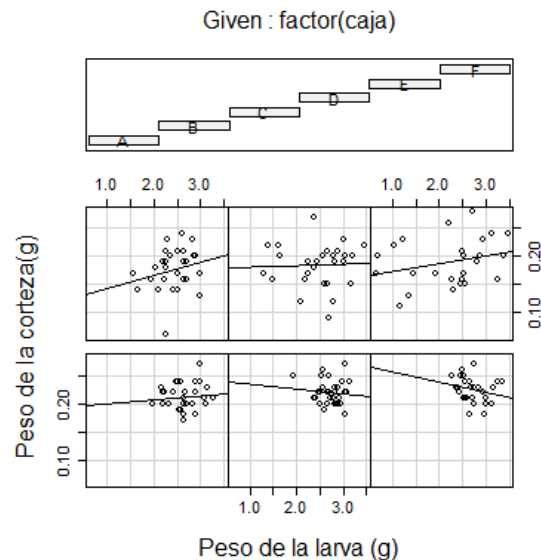


Figura 2. 12: gráfico de coplot que relaciona el peso de la corteza y de las larvas (expresados en g/gusano de seda) para cada una de las seis cajas.

# Este gráfico muestra el "Peso de la corteza" en función del "Peso de la larva" dada una tercera variable, en este caso las "cajas".

# En consecuencia, el panel muestra un gráfico para cada caja siguiendo el orden del panel superior.

# Líneas de regresión NO paralelas indican interacción entre el "Peso de la larva" y las "cajas".

# Este gráfico permite identificar relaciones considerando tres variables. Es exploratorio.

# Combinamos "línea genética" y "genero"

```
with(gusa, coplot(peso_corteza ~ peso_larva|linea * genero, show.given=T,
  ylab = "Peso de la corteza(g)",
  xlab = "Peso de la larva (g)",
  panel = panel.linea))
```

```
with(gusa, coplot(peso_corteza ~ peso_cap|linea * genero, show.given=T,
  ylab = "peso_corteza(g)",
  xlab = "peso del capullo (g)",
  panel = panel.linea))
```

# Recordar que panel.linea fue generado más arriba.

# Observamos la líneas de regresión para evaluar potencial interacción.

### # 9- Independencia

# Este supuesto puede ser violado cuando los datos poseen una relación espacial, temporal o filogenética.

# Entonces, necesitamos modelar estas jerarquías:

# la relación espacial, temporal o de parentesco entre los datos que viola el supuesto de independencia.

# **¡Bienvenidos!**

# Estas estructuras de dependencia pueden modelarse de distintas maneras:

# a- usando una covariable (i.e.: pH, temperatura)

# b- usando modelos mixtos que permiten incluir variables aleatorias (i.e: parcela, bloque)

# c- incluyendo la estructura de correlación mediante modelos gls (generalized least square)

# **IMPORTANTE:** cualquiera sea la forma que utilicemos,

# los residuales del modelo no deben mostrar estructura de dependencia

# (i.e.: debemos observar una nube de puntos sin un patrón).

# Durante el curso vamos a detenernos en el uso de modelos mixtos

# y estructuras de correlación (los casos b y c).

### # 10- Resumen

# Explorar los datos es un paso fundamental para comenzar un análisis.

# En esta práctica hemos visto algunas herramientas para explorar los datos de manera gráfica y analítica.

# Hemos visto alternativas para evaluar multicolinealidad así como algunas sugerencias para evitarla.

# Finalmente, hicimos algunas consideraciones respecto del supuesto de independencia sobre el cual profundizaremos durante las siguientes prácticas.

### # 11- Agradecimiento

Datos gentilmente provistos por Artave Gabriela, Dobler Samanta, Lopez Zieher Ximena María del Laboratorio de sericultura de la Cátedra de Producciones Alternativas de la Facultad de Agronomía de la Universidad de Buenos Aires. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.

# Modelos y jerarquías

---

## Contenidos

<b># CAPÍTULO 3: Modelos y jerarquías</b> .....	<b>33</b>
<b># 1- Configuración inicial</b> .....	<b>33</b>
# 1.a- Definir el directorio de trabajo .....	33
# 1.b- Leer el archivo "gusanos.csv".....	33
<b># 2- El caso: Producción de seda</b> .....	<b>33</b>
# 2.a- Pregunta de interés .....	33
# 2.b- Variable respuesta.....	34
# 2.c- Variables predictoras.....	34
<b># 3- Identificar jerarquías</b> .....	<b>34</b>
# ¿Qué jerarquías podemos identificar en el caso de estudio? .....	34
<b># 4- Plantear modelos</b> .....	<b>34</b>
<b># 5- Modelos de efectos mixtos</b> .....	<b>34</b>
# 5.a- Paquete lme4.....	35
# 5.b- Paquete nlme .....	36
# 5.c- Diferencias entre lme4 y nlme .....	36
<b># 6- Validación del modelo</b> .....	<b>36</b>
# 6.a- Residuales.....	36
# 6.b- Ajustados .....	36
# 6.c- Gráficos .....	36
<b># 7- Comparar los tres modelos</b> .....	<b>37</b>
# 7.a- summary de la función lme .....	37
# 7.b- summary de la función lmer .....	38
# 7.c- summary de la función gls .....	39
<b># 8- Resumen: ¿Qué hicimos hasta aquí?</b> .....	<b>40</b>
<b># 9- Agradecimiento</b> .....	<b>41</b>

---



---



## # CAPÍTULO 3: Modelos y jerarquías

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo "gusanos.csv"

```
# Aclaración: abrir el archivo "gusanos.csv" que generamos en el CAPÍTULO 1
```

```
# gusa<-.....
```

```
# (Utilizar alguna de las maneras vistas en las prácticas anteriores para abrir la tabla "gusanos.csv" que generamos en el capítulo 1)
```

```
summary(gusa)
```

### # 2- El caso: Producción de seda

El objetivo del trabajo fue caracterizar líneas de gusano de seda de origen japonés y chino (E oro y E CH, respectivamente), endocriadas en la FAUBA durante 4 generaciones para evaluar si pueden ser utilizadas en la formación de híbridos comerciales adaptados a la región pampeana. Los gusanos de seda de las líneas E oro y E CH crecieron bajo condiciones controladas de luz, fotoperiodo, humedad y alimentación. Los gusanos fueron colocados en cajas separadas por líneas (30 gusanos por caja en 3 cajas por línea. Fig. 3.1). Ver detalles en el capítulo 2".

¿Qué características de los gusanos podrían predecir la producción de seda?

$2 \text{ líneas} \times 3 \text{ cajas.línea}^{-1} \times 30 \text{ gusanos.cajas}^{-1} = 180 \text{ gusanos}$

Líneas = **EC** (chino) y **Eoro** (japonés)

Ultimo estadio

- Sexo
- Peso larva (g)
- Peso capullo (g)
- Peso pupa (g)
- Peso corteza (g): capullo – pupa
- Seda bruta (%):  $\text{corteza/capullo} \times 100$

Entonces, el **peso de la corteza** estima la **producción de seda**.

El muestreo quedó desbalanceado debido a la muerte de individuos de la línea Eoro

Figura 3. 1: esquema del diseño experimental que evalúa la producción de seda de dos líneas de gusanos de seda.

### # 2.a- Pregunta de interés

# ¿Qué características de los gusanos podrían predecir la producción de seda?

### # 2.b- Variable respuesta

# Peso de la corteza (estimador de la producción de seda).

# La corteza es la parte externa del capullo y se calcula como "capullo - pupa = corteza"

### # 2.c- Variables predictoras

# - línea: línea genética de los gusanos de seda

# - genero

# - peso larva (gramos)

# - peso capullo (gramos)

# - peso pupa (gramos)

### # 3- Identificar jerarquías

# De manera frecuente, los datos con los que trabajamos violan el supuesto de independencia.

# Es decir, los datos poseen una relación espacial, temporal o filogenética.

# Entonces, cuentan con estructura jerárquica (multi-nivel o anidada).

### # ¿Qué jerarquías podemos identificar en el caso de estudio?

# Gusanos contenidos en Cajas

# Es decir, dos niveles

# En este caso, la variable respuesta y las variables predictoras están determinadas a nivel de gusanos.

# Sin embargo, en modelos más complejos podemos encontrar variables medidas en distintos niveles.

# Por ejemplo, mientras el peso esta medido a nivel de gusano

# (tenemos un dato por gusano), la intensidad de la luz que llega puede ser medido a nivel de caja (tenemos un dato por caja).

### # 4- Plantear modelos

```
mod_base<-lm(peso_corteza~linea+genero, data=gusa)
```

# en este caso, "mod\_base" es el nombre del objeto que contiene el modelo

# Empezamos por una función que ajusta modelos lineales (lm)

# Los argumentos de la función son "respuesta" ~ (ñuflo indica "depende de") "predictor1" + "predictor2" ...etc, data="tabla de datos"

# En este caso, el peso de la corteza depende de la línea y el genero.

# ¿Cómo se indican interacciones entre predictores?

```
mod_base<-lm(peso_corteza~linea+genero+linea:genero, data=gusa)
```

# ":" indica interacción

# Este último modelo es equivalente a

```
mod_base<-lm(peso_corteza~linea*genero, data=gusa)
```

# El asterisco considera los efectos simples y las interacciones entre las variables.

### # 5- Modelos de efectos mixtos

# Hasta aquí definimos un modelo lineal de efectos fijos (lm).

# Es decir, ignoramos la estructura de dependencia dada por las cajas que contienen a los gusanos.

# En este caso, la estructura de dependencia está dada por las cajas que contienen a los gusanos.

# Hay varias formas de representar de manera matemática un modelo. A lo largo del curso utilizaremos el siguiente modo que combina regresiones para cada una de las escalas o jerarquías ("Combining separate local regressions").

# Por ejemplo, para el modelo planteado de los gusanos de seda, el modelo se puede representar como:

$$Y_i \sim N(\mu_i, \sigma_\epsilon^2)$$

$$\mu_i = \beta_{0j[i]} + \beta_1 * sexo_i + \beta_2 * sexo_i * linea_j$$

$$\beta_{0j} \sim N(\mu_{\beta_0}, \sigma_{\beta_0}^2)$$

$$\mu_{\beta_0} = \alpha_{0j} + \alpha_1 * linea_j$$

$i = 1, 2, 3, \dots, 180$  gusanos

$j = 1, 2, 3, \dots, 6$  cajas

### Referencias:

$Y_i$  = peso de la corteza

Coefficientes:

$\beta_{0j}$  = intercepción

$\beta_1$  = sexo

$\beta_2$  = sexo \* linea

$\alpha_{0j}$  = intercepción

$\alpha_1$  = linea

$\sigma_{\beta_0}^2$  = varianza entre cajas

$\sigma_\epsilon^2$  = varianza residual

# Podemos incluir la estructura de dependencia (gusanos dentro de cajas) en el modelo mediante el uso de modelos de efectos mixtos.

# Existen al menos tres funciones que permiten trabajar con modelos de efectos mixtos.

### # 5.a- Paquete lme4

library("lme4")

# El paquete lme4 contiene la función "lmer" para ajustar modelos

# con estructuras de dependencia.

```
mod_lmer <- lmer(peso_corteza ~ linea * genero + (1|caja), data=gusa)
```

# "caja" debe ser factor.

# Si utilizamos números para identificarlas, podemos especificar

```
"(1|factor(gusa$caja))"
```

# En este caso, la estructura de dependencia o, factor aleatorio, se indica entre paréntesis como (1|caja)

# Este argumento indica que el modelo ajustará una

# intercepción (ordenada al origen) general e intercepciones distintas para cada caja.

### # 5.b- Paquete nlme

```
library("nlme")
```

```
# En el paquete nlme hay, entre otras, dos funciones que permiten ajustar  
modelos con estructuras de dependencia: lme y gls
```

```
mod_lme<-lme(peso_corteza~linea*genero,random=~1|caja, data=gusa)
```

```
# En este caso, la estructura de dependencia se indica con el argumento  
"random=~1|caja"
```

```
mod_gls<-gls(peso_corteza~linea*genero,  
             correlation=corCompSymm(form=~1|caja), data=gusa)
```

```
# En este caso, la estructura de dependencia se indica con el argumento  
"correlation=corCompSymm(form=~1|caja)"
```

```
# Las tres formas especifican modelos análogos
```

### # 5.c- Diferencias entre lme4 y nlme

```
# Si bien son muy parecidos poseen algunas diferencias:
```

```
# • lme4 puede ser más eficiente en el uso de memoria que nlme.
```

```
# • lme4 permite incluir factores aleatorios cruzados,  
# mientras que esto no es posible en nlme.
```

```
# • lme4 permite ajustar modelos de efectos mixtos lineales generalizados  
(GLMM), a través de la función glmer.
```

```
# Es decir, se pueden ajustar datos con distribuciones de errores distintas de la  
normal como binomial o poisson.
```

```
# Mientras que nlme sólo permite distribución normal.
```

```
# • nlme permite incluir funciones para modelar heterocedasticidad y  
# correlación (temporal, espacial y filogenética) de los residuos.
```

```
# Estas funciones no pueden ser incluidas en lme4.
```

```
# • nlme es más flexible que lme4 para componer estructuras complejas de  
varianzas-covarianzas.
```

```
# • nlme está mejor documentado que lme4 (hasta el momento).
```

## # 6- Validación del modelo

### # 6.a- Residuales

```
E_lmer<-resid(mod_lmer, scaled=TRUE)
```

```
# En lmer le pedimos los residuales "en escala" de manera que
```

```
# considere la estructura de dependencia (el factor aleatorio).
```

```
# El equivalente para el modelo lme sería:
```

```
E_lme<-resid(mod_lme, type="normalized")
```

```
# Con type="normalized", le indicamos que calcule los residuos estandarizados
```

```
# Estos residuos son los que debemos utilizar para validar los modelos de  
efectos mixtos.
```

### # 6.b- Ajustados

```
F_lmer<-fitted(mod_lmer)
```

### # 6.c- Gráficos

```
layout(matrix(1:3, 1,3))
```

```
# la función "layout" permite dimensionar la cantidad de gráficos y distribución
en la ventana de gráficos
plot(x=F_lmer, y=E_lmer, xlab="ajustados", ylab="residuales normalizados")
abline(0,0, col="red", lwd= 3)
boxplot(E_lmer~gusa$linea, main="línea genética")
boxplot(E_lmer~gusa$genero, main="género")
```

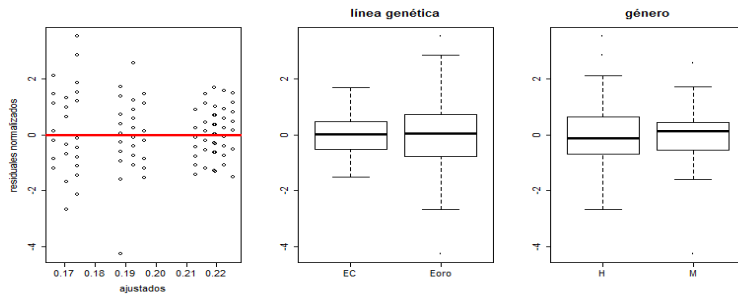


Figura 3. 2: residuales normalizados del modelo "mod\_lmer" en relación a los valores ajustados por el modelo (izq.), de las líneas genéticas (centro) y del género (der.).

# ¡**IMPORTANTE!** Siempre deseamos un gráfico que no muestre un patrón o una tendencia.

# En cambio, en este caso, observamos tres grandes grupos

# con aparentes diferencias en la dispersión de los residuales a lo largo de los valores ajustados.

# Observamos que la línea Eoro y las hembras tienen mayor varianza.

# Más adelante veremos cómo modelar este aspecto.

### # 7- Comparar los tres modelos

```
summary(mod_lme)
```

```
summary(mod_lmer)
```

```
summary(mod_gls)
```

#### # 7.a- summary de la función lme

```
summary(mod_lme)
```

```
# Linear mixed-effects model fit by REML
```

```
# Data: gusa
```

```
# AIC BIC logLik
```

```
# -669.2444 -650.5006 340.6222
```

# Son distintos índices de la bondad de ajuste o verosimilitud del modelo.

# Más adelante usaremos el criterio AIC para seleccionar modelos.

```
# Random effects:
```

```
# Formula: ~1 | caja
```

```
# (Intercept) Residual
```

```
# StdDev: 0.005260823 0.03024337
```

# Observamos el desvío del factor aleatorio (cajas) y de los residuales del modelo.

```
# varianza = desvio^2
```

# **IMPORTANTE:** cuando usemos la función lmer, el modelo nos dará ambos, el desvío y la varianza

```
# Fixed effects: peso_corteza ~ linea * genero
#
#              Value      Std.Error  DF    t-value  p-value
# (Intercept)  0.22235107  0.005453594  164   40.77148  0.0000
# lineaEoro    -0.05210470  0.008119775    4   -6.41701  0.0030
# generoM      -0.00604792  0.006465802  164   -0.93537  0.3510
# lineaEoro:generoM 0.02831680  0.009366297  164    3.02327  0.0029
```

# Observar el modelo genera para los efectos fijos (simples e interacciones) la estimación del parámetro.

# La primera fila hace referencia a la intercepción del modelo. Es decir, el modelo estima que la ordenada al origen es de 0.22.

# **ATENCIÓN:**

# En este caso, el modelo estima el peso de la corteza de pupas Hembras de la línea EC es 0.22

# El peso de la corteza de las Hembras de la línea Eoro es 0.052 menor que las de la línea EC ( $0.22 - 0.052 = 1.68$ )

# El peso de la corteza de los Machos EC es 0.006 menor que las Hembras de la misma línea ( $0.222 - 0.0064 = 0.2156$ )

# y, el peso de la corteza de los Machos Eoro es 0.028 mayor que las Hembras EC ( $0.22 + 0.028 = 0.248$ )

# Luego, las columnas indican el error estándar, los grados de libertad (DF), el valor del estadístico T y el valor de probabilidad asociado.

# **ATENCIÓN:** las pruebas de T comparan los niveles de los factores respecto de la ordenada al origen. Más adelante veremos otro tipo de análisis.

# Correlation:

```
#              (Intr) lineEr generoM
# lineaEoro    -0.672
# generoM      -0.587  0.394
# lineaEoro:generoM  0.405 -0.628 -0.690
```

# Esta tabla de correlación nos hubiera puesto en evidencia multicolinealidad.

# Standardized Within-Group Residuals:

```
#  Min      Q1      Med      Q3      Max
# -4.24827208 -0.63482939  0.01851832  0.58041960  3.50352726
```

#

# Number of Observations: 172

# Number of Groups: 6

# **7.b- summary de la función lmer**

```
summary(mod_lmer)
```

```
# Linear mixed model fit by REML ['lmerMod']
```

```
# Formula: peso_corteza ~ linea * genero + (1 | caja)
```

```

# Data: gusa
#
# REML criterion at convergence: -681.2

# Indica el ajuste del modelo.
# Más adelante usaremos el criterio AIC para seleccionar modelos.

# Scaled residuals:
#   Min       1Q       Median       3Q      Max
# -4.2483  -0.6348   0.0185    0.5804   3.5035
#
# Random effects:
#   Groups             Name             Variance  Std.Dev.
#   caja              (Intercept)  2.768e-05  0.005261
#                   Residual    9.147e-04  0.030243
# Number of obs: 172, groups: caja, 6

# Es equivalente a la información para el modelo lme solo
# que con un orden levemente distinto.
# Observamos la varianza y el desvío del factor aleatorio (cajas) y de los
# residuales del modelo.
# varianza = desvio^2
# IMPORTANTE: cuando usemos la función lme, el modelo nos dará sólo el
# desvío

# Fixed effects:
#               Estimate Std. Error  t value
# (Intercept)   0.222351   0.005454  40.77
# lineaEoro    -0.052105   0.008120  -6.42
# generoM      -0.006048   0.006466  -0.94
# lineaEoro:generoM  0.028317   0.009366   3.02
#
# Correlation of Fixed Effects:
#              (Intr) lineEr generoM
# lineaEoro   -0.672
# generoM     -0.587  0.394
# lineaEr:sxM  0.405 -0.628 -0.690

# 7.c- summary de la función gls
summary(mod_gls)
# Generalized least squares fit by REML
# Model: peso_corteza ~ linea * genero
# Data: gusa
# AIC          BIC          logLik
# -669.2444   -650.5006   340.6222
# El modelo ajustado por mínimos cuadrados, generó valores de ajuste iguales al
# modelo mixto ajustado con la función nlme.

# Comparar:

```

```

# Linear mixed-effects model fit by REML
# Data: gusa
#   AIC      BIC      logLik
# -669.2444 -650.5006  340.6222

# Correlation Structure: Compound symmetry
# Formula: ~1 | caja
# Parameter estimate(s):
# Rho
# 0.02936948

# Cuando usamos esta estructura de correlación, el parámetro "rho"
# es el coeficiente de correlación entre dos residuales de una misma caja (en este
# caso).
# Tanto la intercepción aleatoria utilizada en "mod_lme" como la
# estructura de correlación utilizada en "mod_gls" asumen que la
# correlación entre cualquier par de observaciones dentro de una caja es la
# misma.
# ¿Son similares?

# Coefficients:
#               Value      Std.Error    t-value    p-value
# (Intercept)  0.22235107  0.005453585  40.77154  0.0000
# lineaEoro    -0.05210470  0.008119763  -6.41702  0.0000
# generoM      -0.00604792  0.006465802  -0.93537  0.3509
# lineaEoro:generoM  0.02831679  0.009366297   3.02326  0.0029

# Observar que los valores estimados, los errores estándar, los valores de T y de
# probabilidad son muy similares entre las tres funciones que utilizamos.

# Correlation:
#           (Intr)  lineEr  generoM
# lineaEoro -0.672
# generoM    -0.587  0.394
# lineaEoro:generoM  0.405 -0.628 -0.690
#
# Standardized residuals:
#   Min      Q1      Med      Q3      Max
# -4.31680673 -0.65954423 -0.07658826  0.57493026  3.57532615
#
# Residual standard error: 0.03069752
# Degrees of freedom: 172 total; 168 residual

```

### # 8- Resumen: ¿Qué hicimos hasta aquí?

- # 1. Utilizamos un modelo de efectos mixtos para incluir la estructura de dependencia (dada por las cajas donde crecieron los gusanos).
- # 2. Evaluamos el modelo de manera gráfica: estimamos los residuales normalizados y los valores ajustados por el modelo, observamos la distribución



de los residuales, evaluamos homogeneidad en las varianzas, validamos (o no) el modelo.

# 3. Usamos la función summary: índices de bondad de ajuste, desvío estándar (aleatorio y residual), estimación de los efectos fijos y la correlación entre ellos.

# Seguiremos trabajando sobre este caso en el próximo capítulo.

### # 9- Agradecimiento

Datos gentilmente provistos por Artave Gabriela, Dobler Samanta, Lopez Zieher Ximena María del Laboratorio de sericultura de la Cátedra de Producciones Alternativas de la Facultad de Agronomía de la Universidad de Buenos Aires. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.

Datos jerárquicos

# Estimación y selección de modelos

---

## Contenidos

<b># CAPÍTULO 4: Estimación y selección de modelos</b> .....	<b>43</b>
<b># 1- Configuración inicial</b> .....	<b>43</b>
# 1.a- Definir el directorio de trabajo .....	43
# 1.b- Leer el archivo de datos .....	43
<b># 2- El caso: Producción de seda</b> .....	<b>43</b>
# 2.a- Pregunta de interés .....	43
# 2.b- Variable respuesta.....	44
# 2.c- Variables predictoras.....	44
<b># 3- Identificar jerarquías</b> .....	<b>44</b>
<b># 4- Alternativas de análisis</b> .....	<b>44</b>
# 4.a- "no.pooling" .....	44
# 4.b- "complete.pooling" .....	45
# 4.c- "partial.pooling" o, modelo mixto.....	46
<b># 5- Construcción del modelo</b> .....	<b>46</b>
# 5.a- Modelo partial-pooling.....	47
# 5.b- Ajustamos el componente aleatorio.....	47
# 5.c- Ajustamos el componente fijo .....	48
# 5.d - Tabla resumen del LRT.....	51
# 5.e- Modelo final .....	53
<b># 6- Gráficos de ajuste del modelo</b> .....	<b>54</b>
<b># 7- Presentamos el modelo final</b> .....	<b>55</b>
# 7.a- ¿Qué información nos da el modelo?.....	55
# 7.b- Coeficientes del modelo.....	56
# 7.c- Intervalos de confianza de las estimaciones .....	57
# 7.d- Otros elementos del diseño del modelo.....	57
<b># 8- ANOVA</b> .....	<b>58</b>
<b># 9- Inferencia multimodelo</b> .....	<b>59</b>
<b># 10- Tarea</b> .....	<b>61</b>
<b># 11- Resumen</b> .....	<b>61</b>
<b># 12- Agradecimiento</b> .....	<b>61</b>

---

---

## # CAPÍTULO 4: Estimación y selección de modelos

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo de datos

```
# En este caso, el archivo "gusanos.csv"
```

```
# gusa<-.....
```

```
# (Utilizar alguna de las maneras vistas en los capítulos anteriores para
```

```
# abrir la tabla "gusanos.csv" que generamos en el capítulo 1)
```

```
summary(gusa)
```

### # 2- El caso: Producción de seda

El objetivo del trabajo fue caracterizar líneas de gusano de seda de origen japonés y chino (E oro y E CH, respectivamente), endocriadas en la FAUBA durante 4 generaciones para evaluar si pueden ser utilizadas en la formación de híbridos comerciales adaptados a la región pampeana. Los gusanos de seda de las líneas E oro y E CH crecieron bajo condiciones controladas de luz, fotoperiodo, humedad y alimentación. Los gusanos fueron colocados en cajas separadas por líneas (30 gusanos por caja en 3 cajas por línea. Fig. 4.1). Ver detalles en el "Capítulo 2".



Figura 4. 1: esquema del diseño experimental que evalúa la producción de seda de dos líneas de gusanos de seda.

### # 2.a- Pregunta de interés

# ¿Qué características de los gusanos podrían predecir la producción de seda?

**# 2.b- Variable respuesta**

# Peso de la corteza (estimador de la producción de seda).

# La corteza es la parte externa del capullo y se calcula como "capullo - pupa = corteza"

**# 2.c- Variables predictoras**

# - línea: línea genética de los gusanos de seda

# - genero

# - peso larva (gramos)

# - peso capullo (gramos)

# - peso pupa (gramos)

# En el capítulo anterior observamos colinealidad entre peso\_pupa y peso\_capullo.

# Por lo tanto, utilizaremos peso\_pupa

**# 3- Identificar jerarquías**

# Gusanos contenidos en Cajas

**# 4- Alternativas de análisis**

# Estrictamente, las cajas son independientes, pero no los gusanos dentro de una caja.

# ¿Cuáles son las alternativas para lidiar con esta estructura jerárquica?

# a. "no.pooling": la caja como unidad de análisis.

# b. "complete.pooling": gusano como unidad de análisis.

# c. "partial.pooling" / modelo mixto: gusanos como unidad de análisis teniendo en cuenta el anidamiento (la caja)

**# 4.a- "no.pooling"**

# Las cajas como unidad de análisis.

# Estimamos los promedios para ajustar el modelo con la función "aggregate".

```
?aggregate
```

```
gusa_mean<-with(gusa,
  aggregate(list(peso_pupa, peso_larva,peso_cap,peso_corteza,seda_bruta),
    by=list(caja), mean))
```

```
colnames(gusa_mean)<-c("caja","peso_pupa","peso_larva",
"peso_cap","peso_corteza", "seda_bruta")
```

```
linea<-rep(c("EC", "Eoro"),each=3)
```

```
linea
```

```
gusa_mean<-cbind(linea, gusa_mean)
```

```
gusa_mean
```

	linea	caja	peso_pupa	peso_larva	peso_cap	peso_corteza	seda_bruta
1	EC	A	0.8483333	2.649333	1.060667	0.2123333	20.17467
2	EC	B	0.8376667	2.735	1.058	0.2203333	20.956
3	EC	C	0.877931	2.72931	1.103448	0.2255172	20.64
4	Eoro	D	0.9282759	2.421379	1.104483	0.1762069	15.90724
5	Eoro	E	1.05	2.485	1.234286	0.1842857	14.94179
6	Eoro	F	1.0123077	2.246538	1.202308	0.19	15.75077

```

# Planteamos el modelo:
n.p<-lm(peso_corteza~linea+peso_larva+peso_pupa, data=gusa_mean)
summary(n.p)

Call:
lm(formula = peso_corteza ~ linea + peso_larva + peso_pupa, data =
gusa_mean)

Residuals:
    1         2         3         4         5         6
-7.431e-03  3.115e-03  4.316e-03 -2.003e-05 -2.552e-03  2.572e-03

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.18490    0.11590   1.595  0.2517
lineaEoro    -0.05529    0.01742  -3.173  0.0866 .
peso_larva   -0.01772    0.03719  -0.476  0.6808
peso_pupa     0.09643    0.07496   1.286  0.3271
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.006953 on 2 degrees of freedom
Multiple R-squared:  0.9543, Adjusted R-squared:  0.8858
F-statistic: 13.93 on 3 and 2 DF, p-value: 0.06771

# ¿Qué pasa si nos interesa incluir más variables o interacciones en este modelo?
# ¡NO alcanzan los gados de libertad!

# Una manera alternativa de "no.pooling" sería hacer un modelo independiente
para cada caja.
# En este caso en particular, no hubiéramos podido evaluar las líneas dado que
fueron asignadas a nivel de cajas.

# 4.b- "complete.pooling"
# Los gusanos como unidad de análisis
# Ahora podríamos incluir las interacciones
# ATENCIÓN: es incorrecto ¡Violamos el supuesto de independencia!

c.p<-lm(peso_corteza~linea*genero*peso_pupa*peso_larva, data=gusa)

# ¿Qué pasa si no consideramos la estructura de los datos?
summary(c.p)
# En este análisis estamos cometiendo "pseudo-replicación".
# Una consecuencia de las pseudorréplicas es que el cuadrado medio del error
(CME) estaría estimado con un "n" mayor al real en el denominador (n =
gusanos).
# Entonces, sobre-estima el valor del estadístico (t o F) y el valor de probabilidad
(p-valor) sería menor al real.

# Otras consecuencias de utilizar un "n" erróneo:
# - cambian los valores estimados para los efectos fijos,
# sobre todo cuando el número de observaciones varía de manera importante
entre grupos (cajas).

```

```
# La falta de independencia puede introducir un sesgo importante en las
estimaciones de los efectos fijos,
# (necesitamos el modelo mixto para estimar si esta estructura de dependencia
es importante o no).
# - cambian los errores estándar asociados a cada efecto fijo
# y no necesariamente son más chicos de lo que debieran ser.
```

#### # 4.c- "partial.pooling" o, modelo mixto

```
# Los gusanos como unidad de análisis teniendo en cuenta el anidamiento o
estructura de dependencia (la caja).
```

```
library("nlme")
p.p<-lme(peso_corteza~linea*genero*peso_pupa*peso_larva,
        random=~1|caja, data=gusa)
```

```
# Este modelo permite:
# - modelar tanto las tendencias centrales como la dispersión de los datos
(variación dentro y entre las cajas).
# - modelar respuestas a distinta escala y su interacción.
# - modelar la dinámica espacial y temporal.
# - más precisión (menos varianza residual).
# - más eficiente en el uso de los grados de libertad.
# - las estimaciones consideran (ponderan) el tamaño de la muestra dentro del
grupo (la caja) y la variación dentro y entre los grupos (cajas)
# Es un caso intermedio, los estimados estarán más cerca del modelo complete-
pooling para las cajas con menor observaciones y,
# más cerca del no-pooling para las caja con más observaciones.
```

#### # 5- Construcción del modelo

```
# a. Partiendo de nuestro modelo partial-pooling (o modelo mixto) que
# contempla la estructura dada por el diseño del experimento y el método de
muestreo.
# Escribimos el modelo con tantos factores e interacciones
# que uno considere importantes.

# b. Ajustamos el componente aleatorio del modelo.
# Usamos REML para comparar distintas estructuras aleatorias.
# Verificamos la validez del modelo (el cumplimiento de los supuesto) a partir de
los residuales estandarizados ("normalized").

# c. Ajustamos el componente fijo del modelo con ML.
# Para este punto existen distintos marcos conceptuales y procedimientos
# (p.ej.: tabla de anova del modelo, cocientes de verosimilitud (LRT), wald test, t-
Student, etc.).
# Cualquiera de estos procedimientos es correcto si se lo utiliza adecuadamente.
# A lo largo del curso aplicaremos varios de ellos.
# d. Presentamos el modelo final utilizando el método REML de estimación
```

**# 5.a- Modelo partial-pooling**

# Recordemos, el modelo que planteamos anteriormente: caja como intercepción aleatoria

```
p.p<-lme(peso_corteza~linea*genero*peso_pupa*peso_larva,
        random=~1|caja, data=gusa)
```

# Pregunta: ¿Una variable cuantitativa podría ser una intercepción aleatoria?

# Observemos la salida de un modelo ajustado con la función "lme"

```
summary(p.p)
```

```
# Linear mixed-effects model fit by REML
```

```
# Data: gusa
```

```
# AIC BIC logLik
```

```
# -697.9192 -643.0218 366.9596
```

```
#
```

```
# Random effects:
```

```
# Formula: ~1 | caja
```

```
# (Intercept) Residual
```

```
# StdDev: 0.008124325 0.02195615
```

# El primer valor (sd1)<sup>2</sup> estima la varianza para el término aleatorio

# (variación entre cajas).

# El segundo valor (sd2)<sup>2</sup> estima la varianza residual del modelo

# (variación entre gusanos dentro de cada caja).

# La correlación entre observaciones dentro de un mismo grupo está dado por

```
# (sd1)^2/((sd1)^2+(sd2)^2)= 0.12
```

```
0.008124325^2/(0.008124325^2+0.02195615^2)
```

# Cuanto mayor sea el cociente, mayor es la proporción de la varianza que está siendo controlada por el factor aleatorio.

# Además, este cociente representa la correlación entre dos gusanos que pertenecen a la misma caja.

# Recordamos que siempre es variación no explicada por la componente fija del modelo.

**# 5.b- Ajustamos el componente aleatorio**

# Validación del modelo

# Homogeneidad en las varianzas

# Ajustados vs. residuales

```
E_lme<-resid(p.p, type="normalized")
```

```
F_lme<-fitted(p.p)
```

```
layout(matrix(1:6, 2,3))
```

```
plot(x=F_lme, y=E_lme, xlab="ajustados", ylab="residuales estandarizados")
```

```
abline(0,0,col="red", lwd=3)
```

# Cada una de las variables vs. residuales

```
boxplot(E_lme~gusa$linea, main="linea", ylab="residuales estandarizados")
```

```
boxplot(E_lme~gusa$genero, main="genero", ylab="residuales estandarizados")
```

```
plot(gusa$peso_larva, E_lme, main="larva", ylab="residuales estandarizados")
```

```
abline(0,0,col="red", lwd=3)
plot(gusa$peso_pupa, E_lme, main="pupa", ylab="residuales estandarizados")
abline(0,0,col="red", lwd=3)
```

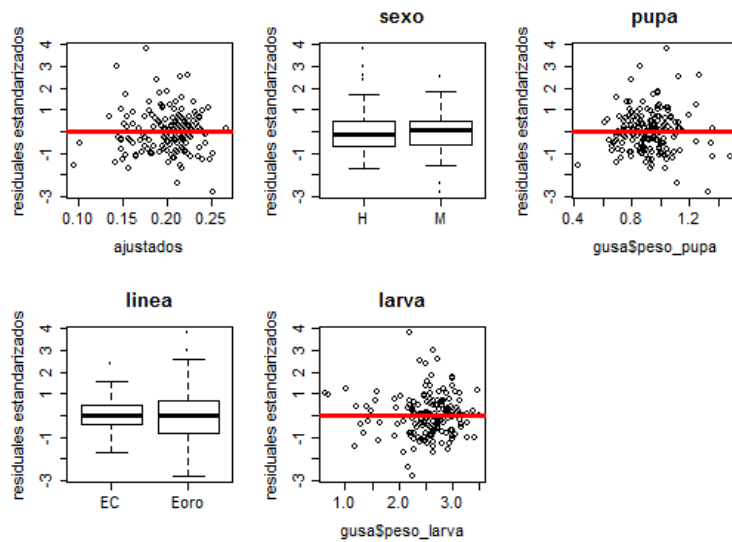


Figura 4. 2: residuales normalizados del modelo "p.p" en relación a los valores ajustados por el modelo y cada uno de los predictores (genero de los gusanos, peso de las pupas, linea genética y peso de las larvas).

```
# Observamos patrones para las variables "linea" (y "peso_larva").
# Estos patrones indican heterogeneidad en las varianzas.
# En los próximos capítulos veremos algunas herramientas para modelar las
varianzas.
# Una vez modeladas las varianzas, tenemos la estructura óptima para el
componente aleatorio del modelo.
```

#### # 5.c- Ajustamos el componente fijo

```
# ¡IMPORTANTE! ahora debemos ajustar el modelo por máxima verosimilitud
(ML).
# Para esto utilizamos la función "update"
p.p1<-update(p.p, method="ML")
```

```
# Recordamos: para este punto existen distintos marcos conceptuales y
procedimientos (p.ej.: tabla de anova del modelo, cocientes de verosimilitud
(LRT), wald test, t-Student, etc.). Cualquiera de estos procedimientos es
# correcto si se lo utiliza adecuadamente.
```

```
# Utilizaremos los cocientes de verosimilitud (LRT).
# Este procedimiento consiste en comparar el modelos anidados.
# Es decir, partiendo del modelo completo (p.p1) ir sacando los factores de a uno
y comparando el ajuste de ambos modelos.
# Las comparaciones se hacen por nivel de complejidad en las interacciones
desde las más complejas hasta los factores simples.
```

#### # 5.c.1- Revisar los efectos fijos

```
summary(p.p1)$tTable
```



**# 5.c.2- Comenzar sacando la interacción más compleja: cuádruple.**

# Para esto utilizamos la función "update"

```
p.p2<-update(p.p1, ~.-linea:genero:peso_pupa:peso_larva)
```

**# 5.c.3- Comparar los dos modelos**

```
anova(p.p1, p.p2)
```

# El LRT indica que no hay diferencias en el ajuste del modelo con o sin la interacción.

# Por parsimonia, excluirémos la interacción del modelo.

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p1	1	18	-797.11	-740.45	416.55			
p.p2	2	17	-799.11	-745.60	416.55	1 vs 2	0.0004056552	0.9839

# Dado que no hay otra interacción cuádruple,

# la sacamos del modelo y continuamos sacando las triples de a una.

**# 5.c.4- Sacamos de a una las interacciones triples y comparamos los modelos**

```
summary(p.p2)$tTable
```

```
p.p3<-update(p.p2, ~.-genero:peso_pupa:peso_larva)
```

```
anova(p.p2, p.p3)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p2	1	17	-799.1096	-745.6022	416.5548			
p.p3	2	16	-800.7866	-750.4267	416.3933	1 vs 2	0.322967	0.5698

```
p.p4<-update(p.p2, ~.-linea:peso_pupa:peso_larva)
```

```
anova(p.p2, p.p4)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p2	1	17	-799.1096	-745.6022	416.5548			
p.p4	2	16	-799.3764	-749.0165	415.6882	1 vs 2	1.733246	0.188

```
p.p5<-update(p.p2, ~.-linea:genero:peso_pupa)
```

```
anova(p.p2, p.p5)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p2	1	17	-799.1096	-745.6022	416.5548			
p.p5	2	16	-800.9745	-750.6146	416.4873	1 vs 2	0.1350901	0.7132

```
p.p6<-update(p.p2, ~.-linea:genero:peso_larva)
```

```
anova(p.p2, p.p6)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p2	1	17	-799.1096	-745.6022	416.5548			
p.p6	2	16	-800.9748	-750.6149	416.4874	1 vs 2	0.1348367	0.7135

# En este caso, no hubo diferencias significativas entre los modelos.

# Por parsimonia, sacamos todas las interacciones triples del modelo.

# Si hubiéramos encontrado diferencias en alguno/s de los casos,

# esas interacciones deberían haber quedado en el nuevo modelo.

**# 5.c.5- Sacamos de a una las interacciones dobles y comparamos los modelos**

# Podríamos seguir con el update, pero escribimos el nuevo modelo para practicar

# manteniendo el método de ML

```
p.p7<-lme(peso_corteza~linea+genero+peso_pupa+peso_larva
          +linea:genero
          +linea:peso_pupa)
```

```
+linea:peso_larva
+genero:peso_pupa
+genero:peso_larva
+peso_pupa:peso_larva,
random=~1|caja, data=gusa, method="ML")
```

```
summary(p.p7)$tTable
```

```
p.p8<-update(p.p7, ~.-peso_pupa:peso_larva)
```

```
anova(p.p7, p.p8)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p7	1	13	-804.5680	-763.6505	415.2840			
p.p8	2	12	-805.6354	-767.8654	414.8177	1 vs 2	0.932596	0.3342

```
p.p9<-update(p.p7, ~.-genero:peso_larva)
```

```
anova(p.p7, p.p9)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p7	1	13	-804.568	-763.6505	415.284			
p.p9	2	12	-805.900	-768.1301	414.950	1 vs 2	0.6679291	0.4138

```
p.p10<-update(p.p7, ~.-genero:peso_pupa)
```

```
anova(p.p7, p.p10)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p7	1	13	-804.5680	-763.6505	415.2840			
p.p10	2	12	-806.5508	-768.7809	415.2754	1 vs 2	0.0171296	0.8959

```
p.p11<-update(p.p7, ~.-linea:peso_larva)
```

```
anova(p.p7, p.p11)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p7	1	13	-804.5680	-763.6505	415.2840			
p.p11	2	12	-806.4352	-768.6653	415.2176	1 vs 2	0.13272	0.7156

```
p.p12<-update(p.p7, ~.-linea:peso_pupa)
```

```
anova(p.p7, p.p12)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p7	1	13	-804.568	-763.6505	415.284			
p.p12	2	12	-806.554	-768.7840	415.277	1 vs 2	0.01398306	0.9059

```
p.p13<-update(p.p7, ~.-linea:genero)
```

```
anova(p.p7, p.p13)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p7	1	13	-804.57	-763.6505	415.2840			
p.p13	2	12	-806.51	-768.7422	415.2561	1 vs 2	0.05579479	0.8133

```
# En este caso, tampoco hubo diferencias entre los modelos.
```

#### *# 5.c.6- Sacamos de a uno los factores simples y comparamos los modelos*

```
# Escribimos el nuevo modelo manteniendo el método de ML
```

```
p.p14<-lme(peso_corteza~linea+genero+peso_pupa+peso_larva,
random=~1|caja, data=gusa, method="ML")
```

```
summary(p.p14)$tTable
```

```
p.p15<-update(p.p14, ~.-peso_larva)
```

```
anova(p.p14, p.p15)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
--	-------	----	-----	-----	--------	------	---------	---------

```
p.p14      1  7 -812.3820 -790.3496 413.191
p.p15      2  6 -809.7079 -790.8230 410.854 1 vs 2 4.674099 0.0306
```

```
p.p16<-update(p.p14, ~.-peso_pupa)
```

```
anova(p.p14, p.p16)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p14	1	7	-812.3820	-790.3496	413.1910			
p.p16	2	6	-697.5996	-678.7146	354.7998	1 vs 2	116.7825	<.0001

```
p.p17<-update(p.p14, ~.-genero)
```

```
anova(p.p14, p.p17)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p14	1	7	-812.3820	-790.3496	413.1910			
p.p17	2	6	-756.6637	-737.7787	384.3318	1 vs 2	57.71835	<.0001

```
p.p18<-update(p.p14, ~.-linea)
```

```
anova(p.p14, p.p18)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
p.p14	1	7	-812.3820	-790.3496	413.1910			
p.p18	2	6	-796.0092	-777.1243	404.0046	1 vs 2	18.3728	<.0001

# En este caso, todos los factores simples son importantes

#### # 5.d - Tabla resumen del LRT

```
Termino<-c("linea:genero:peso_pupa:peso_larva",
  "",
  "genero:peso_pupa:peso_larva",
  "linea:peso_pupa:peso_larva",
  "linea:genero:peso_pupa",
  "linea:genero:peso_larva",
  "",
  "peso_pupa:peso_larva",
  "genero:peso_larva",
  "genero:peso_pupa",
  "linea:peso_larva",
  "linea:peso_pupa",
  "linea:genero",
  "",
  "peso_larva",
  "peso_pupa",
  "genero",
  "linea")
```

```
LRT<-c(round(anova(p.p1, p.p2)[2,"L.Ratio"],4),
  "",
  round(anova(p.p2, p.p3)[2,"L.Ratio"],4),
  round(anova(p.p2, p.p4)[2,"L.Ratio"],4),
  round(anova(p.p2, p.p5)[2,"L.Ratio"],4),
  round(anova(p.p2, p.p6)[2,"L.Ratio"],4),
  "",
  round(anova(p.p7, p.p8)[2,"L.Ratio"],4),
  round(anova(p.p7, p.p9)[2,"L.Ratio"],4),
  round(anova(p.p7, p.p10)[2,"L.Ratio"],4),
```

```

round(anova(p.p7, p.p11)[2,"L.Ratio"],4),
round(anova(p.p7, p.p12)[2,"L.Ratio"],4),
round(anova(p.p7, p.p13)[2,"L.Ratio"],4),
"",
round(anova(p.p14, p.p15)[2,"L.Ratio"],4),
round(anova(p.p14, p.p16)[2,"L.Ratio"],4),
round(anova(p.p14, p.p17)[2,"L.Ratio"],4),
round(anova(p.p14, p.p18)[2,"L.Ratio"],4))

gl<-c(summary(p.p1)$tTable["lineaEoro:generoM:peso_pupa:peso_larva", "DF"],
      "",
      summary(p.p1)$tTable["generoM:peso_pupa:peso_larva", "DF"],
      summary(p.p1)$tTable["lineaEoro:peso_pupa:peso_larva", "DF"],
      summary(p.p1)$tTable["lineaEoro:generoM:peso_pupa", "DF"],
      summary(p.p1)$tTable["lineaEoro:generoM:peso_larva", "DF"],
      "",
      summary(p.p1)$tTable["peso_pupa:peso_larva", "DF"],
      summary(p.p1)$tTable["generoM:peso_larva", "DF"],
      summary(p.p1)$tTable["generoM:peso_pupa", "DF"],
      summary(p.p1)$tTable["lineaEoro:peso_larva", "DF"],
      summary(p.p1)$tTable["lineaEoro:peso_pupa", "DF"],
      summary(p.p1)$tTable["lineaEoro:generoM", "DF"],
      "",
      summary(p.p1)$tTable["peso_larva", "DF"],
      summary(p.p1)$tTable["peso_pupa", "DF"],
      summary(p.p1)$tTable["generoM", "DF"],
      summary(p.p1)$tTable["lineaEoro", "DF"])

p_value<-c(round(anova(p.p1, p.p2)[2,"p-value"],3),
           "",
           round(anova(p.p2, p.p3)[2,"p-value"],3),
           round(anova(p.p2, p.p4)[2,"p-value"],3),
           round(anova(p.p2, p.p5)[2,"p-value"],3),
           round(anova(p.p2, p.p6)[2,"p-value"],3),
           "",
           round(anova(p.p7, p.p8)[2,"p-value"],3),
           round(anova(p.p7, p.p9)[2,"p-value"],3),
           round(anova(p.p7, p.p10)[2,"p-value"],3),
           round(anova(p.p7, p.p11)[2,"p-value"],3),
           round(anova(p.p7, p.p12)[2,"p-value"],3),
           round(anova(p.p7, p.p13)[2,"p-value"],3),
           "",
           round(anova(p.p14, p.p15)[2,"p-value"],3),
           round(anova(p.p14, p.p16)[2,"p-value"],3),
           round(anova(p.p14, p.p17)[2,"p-value"],3),
           round(anova(p.p14, p.p18)[2,"p-value"],3))

Table<-data.frame(Termino,LRT,gl,p_value)

```

Table

	Termino	LRT	gl	p_value
1	línea:genero:peso_pupa:peso_larva	4.00E-04	152	0.984
2				
3	genero:peso_pupa:peso_larva	0.323	152	0.57
4	línea:peso_pupa:peso_larva	1.7332	152	0.188
5	línea:genero:peso_pupa	0.1351	152	0.713
6	línea:genero:peso_larva	0.1348	152	0.713
7				
8	peso_pupa:peso_larva	0.9326	152	0.334
9	genero:peso_larva	0.6679	152	0.414
10	genero:peso_pupa	0.0171	152	0.896
11	línea:peso_larva	0.1327	152	0.716
12	línea:peso_pupa	0.014	152	0.906
13	línea:genero	0.0558	152	0.813
14				
15	peso_larva	4.6741	152	0.031
16	peso_pupa	116.7825	152	0
17	genero	57.7183	152	0
18	línea	18.3728	4	0

**# 5.e- Modelo final**

# El modelo final sería el modelo 14 ajustado por el método de REML  
`p.p19<-lme(peso_corteza~línea+genero+peso_pupa+peso_larva,  
random=~1|caja, data=gusa, method="REML")`

# Este modelo se puede representar de manera matemática como:

$$Y_i \sim N(\mu_i, \sigma_\epsilon^2)$$

$$\mu_i = \beta_{0j[i]} + \beta_1 * sexo_i + \beta_2 * peso\_pupa_i + \beta_3 * peso\_larva_i$$

$$\beta_{0j} \sim N(\mu_{\beta_0}, \sigma_{\beta_0}^2)$$

$$\mu_{\beta_0} = \alpha_{0j} + \alpha_1 * línea_j$$

$i = 1, 2, 3, \dots, 180$  gusanos  
 $j = 1, 2, 3, \dots, 6$  cajas

**Donde,**

$Y_i$  = peso de la corteza

Coeficientes:

$\beta_{0j}$  = intercepción

$\beta_1$  = sexo

$\beta_2$  = peso\_pupa

$\beta_3$  = peso\_larva

$\alpha_{0j}$  = intercepción

$\alpha_1$  = línea

$\sigma_{\beta_0}^2$  = varianza entre cajas

$\sigma_\epsilon^2$  = varianza residual

```
# Una manera alternativa para este marco inferencial es la función "Anova" (con
mayúscula) de la librería "car"
# Esta función devuelve una table con el análisis de deviance de tipo II o III para
un modelo dado.
# Además, de modelos ajustados en lmer y lme4, la función "Anova" admite
modelos ajustados en lm, glm, multinom (en nnet), polr (en MASS), coxph (en
survival).
# Para modelos lineales calcula los estadísticos F-tests;
# Para modelos lineales generalizados, el Chi-cuadrado de la relación de
verosimilitud (likelihood-ratio chisquare, Wald chisquare, or F-tests)
```

```
library(car)
Anova(p.p1, test.statistic="LR")
Analysis of Deviance Table (Type II tests)

Response: peso_corteza
```

	Chisq	Df	Pr(>Chisq)
línea	95.2347	1	< 2e-16 ***
género	67.7071	1	< 2e-16 ***
peso_pupa	163.0723	1	< 2e-16 ***
peso_larva	4.2663	1	0.03888 *
línea:género	0.2010	1	0.65395
línea:peso_pupa	0.0421	1	0.83746
género:peso_pupa	0.0000	1	0.99837
línea:peso_larva	0.1220	1	0.72685
género:peso_larva	0.2212	1	0.63809
peso_pupa:peso_larva	0.4640	1	0.49575
línea:género:peso_pupa	0.1375	1	0.71083
línea:género:peso_larva	0.1369	1	0.71140
línea:peso_pupa:peso_larva	1.7652	1	0.18398
género:peso_pupa:peso_larva	0.3238	1	0.56935
línea:género:peso_pupa:peso_larva	0.0004	1	0.98393

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

```
# Observar que el modelo final resultante es igual al modelo que obtuvimos
haciendo el LRT de manera manual.
```

### # 6- Gráficos de ajuste del modelo

```
# Errores y ajustados
E19<-resid(p.p19, type="normalized")
F19<-fitted(p.p19)

layout(matrix(1:6, 2,3))
# la función "layout" permite dimensionar la cantidad de gráficos
# y distribución en la ventana de gráficos
plot(x=F19, y=E19, xlab="ajustados", ylab="residuales normalizados")
abline(0,0, col="red", lwd= 2)
# ¡IMPORTANTE! Siempre deseamos un gráfico que no muestre un patrón o una
tendencia.
```

```
boxplot(E19~gusa$línea, main="línea genética")
boxplot(E19~gusa$género, main="género")
plot(E19~gusa$peso_pupa, main="peso pupa")
```

```
abline(0,0, col="red", lwd= 2)
plot(E19~gusa$peso_larva, main="peso larva")
abline(0,0, col="red", lwd= 2)
```

```
# Ajustados vs. observados
plot(F19,gusa$peso_corteza, xlim=c(0,0.3),ylim=c(0,0.3), xlab="Observados",
ylab="Ajustados")
abline(0,1,col="red", lwd= 3) # agregamos una línea con ordenada al origen = 0 y
pendiente = 1
```

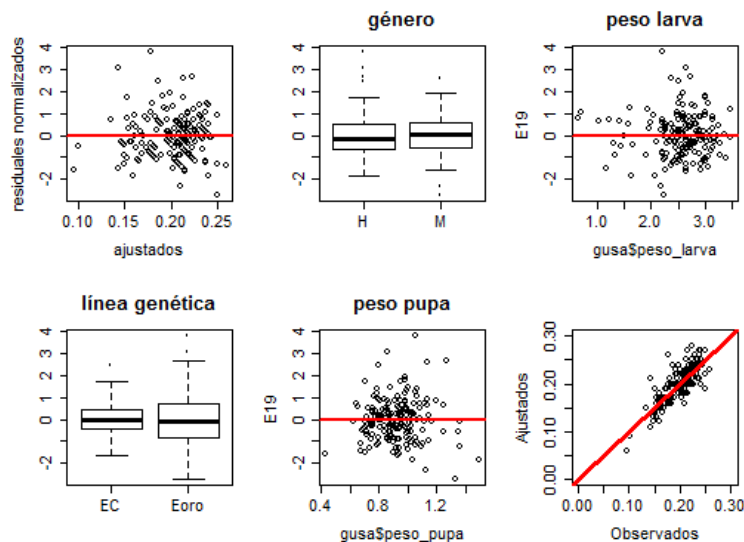


Figura 4. 3: residuales normalizados del modelo "p.p19" en relación a los valores ajustados por el modelo, cada uno de los predictores (género, peso de las pupas, línea genética y peso de las larvas). Valores ajustados por el modelo en relación a los observados en el cuadrante inferior derecho.

## # 7- Presentamos el modelo final

### # 7.a- ¿Qué información nos da el modelo?

```
summary(p.p19)
Linear mixed-effects model fit by REML
Data: gusa
      AIC      BIC    logLik
-768.6362 -746.8102  391.3181

Random effects:
Formula: ~1 | caja
      (Intercept)  Residual
StdDev: 0.007483224 0.02168523

Fixed effects: peso_corteza ~ linea + genero + peso_pupa +
peso_larva
              Value  Std.Error  DF  t-value  p-value
(Intercept)  0.02487024 0.017006622 163  1.462386  0.1456
lineaEoro    -0.06259945 0.007384044   4 -8.477664  0.0011
generoM      0.03354959 0.003941808 163  8.511218  0.0000
peso_pupa    0.18338817 0.014095665 163 13.010254  0.0000
peso_larva   0.00782969 0.003575285 163  2.189950  0.0299
Correlation:
      (Intr) lineEr generoM  pes_pp
lineaEoro -0.047
generoM    -0.443 -0.199
peso_pupa -0.770 -0.296  0.515
peso_larva -0.564  0.157 -0.064  0.005
```

```
Standardized Within-Group Residuals:
      Min       Q1       Med       Q3       Max
-2.73405338 -0.58760375 -0.03384665  0.53912392  3.78651248
```

```
Number of Observations: 172
Number of Groups: 6
```

```
names(summary(p.p19))
[1] "modelStruct" "dims"      "contrasts"  "coefficients" "varFix"
[6] "sigma"       "apVar"     "logLik"     "numIter"      "groups"
[11] "call"        "terms"     "method"     "fitted"       "residuals"
[16] "fixDF"       "na.action" "data"       "corFixed"     "tTable"
[21] "BIC"        "AIC"
```

# Entonces, podemos por ejemplo

```
summary(p.p19)[11]
$call
lme.formula(fixed = peso_corteza ~ linea + genero + peso_pupa +
  peso_larva, data = gusa, random = ~1 | caja, method = "REML")
```

# o también

```
summary(p.p19)$call
lme.formula(fixed = peso_corteza ~ linea + genero + peso_pupa +
  peso_larva, data = gusa, random = ~1 | caja, method = "REML")
```

#### # 7.b- Coeficientes del modelo

```
summary(p.p19)$coefficients
$fixed
  (Intercept)      lineaEoro      generoM      peso_pupa      peso_larva
  0.024870238 -0.062599445  0.033549588  0.183388174  0.007829695

$random
$random$caja
  (Intercept)
A -0.005272052
B  0.006351352
C -0.001079300
D  0.002141694
E -0.007976579
F  0.005834885
```

# Observamos que el modelo estima coeficientes para los factores fijos con una intercepción general

# Además, estima una intercepción para cada una de las cajas (factor aleatorio)

# Entonces, el modelo permite predecir el peso de la corteza general (ponderado por todas las cajas) y,

# el peso de la corteza para cada una de las cajas.

# Si queremos predecir el peso de la corteza de una caja "x" utilizamos el modelo general

# Intercepción

```
a<-round(p.p19$coefficients$fixed[1], 2)
```

# Factores

```
lin<-round(p.p19$coefficients$fixed[2], 2)
```

```
sex<-round(p.p19$coefficients$fixed[3], 2)
```

```
pepu<-round(p.p19$coefficients$fixed[4], 2)
```

```
pelar<-round(p.p19$coefficients$fixed[5], 2)
```



```
# Modelo general
paste("peso corteza = ",a," + ", lin, " xlinea + ", sex, "xgenero + ", pepu,"xpeso pupa
+ ", pelar, "xpeso larva + Error", sep="")
```

```
peso corteza = 0.02 + -0.06 x linea + 0.03x genero + 0.18x peso pupa
+ 0.01x peso larva + Error
```

### # 7.c- Intervalos de confianza de las estimaciones

# La función "intervals" devuelve los intervalos de confianza de las estimaciones de los efectos fijos y aleatorios del modelo.

# Esta función es aplicable a modelos ajustados por lme y gls (librería nlme). Más adelante veremos funciones similares para modelos ajustados lmer (librería lme4)

intervals(p.p19) # el argumento level=0.95, permite definir el nivel de confianza del intervalo.

```
Approximate 95% confidence intervals
Fixed effects:
              lower      est.      upper
(Intercept) -0.0087114558  0.024870238  0.05845193
lineaEoro    -0.0831008391 -0.062599445 -0.04209805
generoM      0.0257659963  0.033549588  0.04133318
peso_pupa    0.1555545275  0.183388174  0.21122182
peso_larva   0.0007698479  0.007829695  0.01488954
attr(,"label")
[1] "Fixed effects:"

Random Effects:
Level: caja
              lower      est.      upper
sd((Intercept)) 0.00300714  0.007483224  0.0186219

within-group standard error:
              lower      est.      upper
0.01945416  0.02168523  0.02417218
```

# De manera alternativa, la función "ci" de la librería "gmodels" también devuelve los intervalos de confianza pero, solo de los efectos fijos del modelo.

```
library(gmodels)
ci(p.p19)
```

### # 7.d- Otros elementos del diseño del modelo

```
library(RLRsim)
```

# La función extract.lmeDesign permite ver varios elementos del diseño de modelos ajustados por lme, mer o lmerMod.

```
names(extract.lmeDesign(p.p19))
```

# • "Vr" calcula la covarianza de los efectos aleatorios dividido por la varianza estimada de los residuos

```
# • "X" Diseño de los efectos fijos
# • "Z" Diseño de los efectos aleatorios
# • "sigmasq" varianza de los residuos
# • "lambda" relación entre las varianzas de los efectos aleatorios y la varianza
de los residuales
# • "y" variable de respuesta
```

```
# Entonces, por ejemplo:
extract.lmeDesign(p.p19)$Vr
```

```
extract.lmeDesign(p.p19)$lambda
[1] 0.1190829
```

```
# Es decir,
# summary(p.p19)
Random effects:
  Formula: ~1 | caja
            (Intercept)  Residual
StdDev: 0.007483224 0.02168523
```

```
0.007483224^2 / 0.02168523^2
```

# Cuanto mayor sea este valor, mayor será la variabilidad no explicada por el modelo entre unidades del factor aleatorio (cajas en este caso), respecto de la variabilidad no explicada por el modelo entre observaciones (a nivel de gusanos en este caso).

### # 8- ANOVA

# En el punto 5.c utilizamos las pruebas de cocientes de verosimilitudes (LTR) # con una secuencia particular de contrastes de modelos.  
# Otro marco inferencial es el de inferencia a partir de un análisis de varianza (devianza) para un modelo, en este caso. Este marco inferencial responde a una filosofía frecuentista.

```
anova(p.p)
```

	numDF	denDF	F-value	p-value
(Intercept)	1	152	2941.5744	<.0001
línea	1	4	23.3772	0.0084
género	1	152	5.1069	0.0253
peso_pupa	1	152	164.9627	<.0001
peso_larva	1	152	4.7021	0.0317
línea:género	1	152	0.7576	0.3855
línea:peso_pupa	1	152	0.2970	0.5866
género:peso_pupa	1	152	0.0045	0.9466
línea:peso_larva	1	152	0.9788	0.3241
género:peso_larva	1	152	1.1899	0.2771
peso_pupa:peso_larva	1	152	0.7255	0.3957
línea:género:peso_pupa	1	152	0.4104	0.5228
línea:género:peso_larva	1	152	0.3128	0.5768
línea:peso_pupa:peso_larva	1	152	1.5534	0.2145
género:peso_pupa:peso_larva	1	152	0.2724	0.6025
línea:género:peso_pupa:peso_larva	1	152	0.0001	0.9922

# La salida muestra los grados de libertad del numerador (NumDF), los grados de libertad del denominador (denDF),

```
# El valor del estadístico F (F-value) y el valor de probabilidad asociado (p-value)
# Observar que el factor "Linea" asociado a las cajas, posee grados de libertad distintos de los demás factores asociados a las pupas.
```

```
# Llegamos a la misma conclusión que con LRT del punto 5...
# En próximos capítulos presentaremos otras opciones de inferencia para debatir.
```

### # 9- Inferencia multimodelo

```
# En el punto 5.c utilizamos las pruebas de cocientes de verosimilitudes (LTR)
# con una secuencia particular de contrastes de modelos.
# Otro marco inferencial es el de inferencia multimodelo a partir de AIC.
```

```
# Generamos múltiples modelos
```

```
library("MuMIn")
```

```
# Partimos del modelo completo de partial pooling ajustado por máxima verosimilitud (ML)
```

```
p.pML<-update(p.p, method="ML")
```

```
p.pML$call
```

```
lme.formula(fixed = peso_corteza ~ linea * genero * peso_pupa *
  peso_larva, data = gusa, random = ~1 | caja, method = "ML")
```

```
sel_1<-dredge(p.pML)
```

```
# IMPORTANTE: la función dredge genera todos los modelos posibles
# (incluido el modelo nulo que solo tiene a la intercepción) y los compara ajustados por ML.
```

```
# Se debe comparar con ML (y no con REML) dado que el efecto aleatorio es el mismo en todos los modelos.
```

```
nrow(sel_1)
```

```
[1] 167
```

```
# Contamos la cantidad de filas que tiene la tabla "sel_1"
```

```
# La función dredge generó 167 modelos
```

```
sel_1
```

```
# Vemos los primeros seis
```

```
head(sel_1)
```

```
Global model call: lme.formula(fixed = peso_corteza ~ linea * genero
  * peso_pupa *
  peso_larva, data = gusa, random = ~1 | caja, method = "ML")
```

```
---
```

```
Model selection table
```

	Int	l	pl	pp	sx	l: pp	l: sx	pl: pp	pl: sx	df	logLik	AICc	dlt	w
144	0.13	+	-0.03	0.08	+			0.04		8	414.7	-812.6	0	0.29
16	0.03	+	0.00	0.18	+					7	413.2	-811.7	0.8	0.19
272	0.01	+	0.01	0.18	+				+	8	414.1	-811.3	1.2	0.16
400	0.11	+	-0.02	0.09	+			0.03	+	9	415.1	-811.2	1.3	0.15
208	0.12	+	-0.03	0.08	+		+	0.04		9	414.9	-810.7	1.8	0.11
176	0.11	+	-0.03	0.09	+	+		0.04		9	414.8	-810.5	2.0	0.10

```
Random terms (all models):
```

```
'1 | caja'
```

Las columnas indican ordenada al origen (Int), línea(1), peso de larvas (pl), peso de pupas (pp), genero (sx), y las interacciones indicadas con ":", grados de libertad (df), verosimilitud (logLik), AICc, delta (dlt) y el peso ponderado (w).

# La salida es una tabla ordenada de los modelos del mejor al peor ajuste según AIC.

# La tabla contiene:

# un número que indentifica al modelo en la primera columna. Este número depende de las iteraciones que hace la función "dredge" y cambia cada vez que corremos la función.

# Es decir, un mismo modelo (tendrá los mismos parámetros estimados y valores de ajuste) puede ser nombrado con distintos números (primera columna) entre corridas.

# la estimación de los predictores continuos

# y un signo "+" para los predictores categóricos incluidos en el modelo.

# Además, muestra la cantidad de parámetros (df), el logLik, el AICc,

# el delta (la diferencia respecto del mejor modelo) y el weight (peso relativo).

# En este caso, hay muchos modelos con relativamente poco peso y bajo delta respecto del primero.

# Es decir, no hay un modelo mucho mejor que el o los siguientes.

# Otra función interesante es "importance"

```
rel_impor<-importance(sel_1)
```

```
rel_impor
```

# La función "importance" calcula para cada variable predictora

# la suma de los pesos relativos de AIC (weights) de los modelos que incluyen

# a la variable.

```
layout(matrix(1:1,1,1))
```

```
barplot(rel_impor[1:4], main= "Importancia relativa de las variables", las=2, col=rainbow(15))
```

# La comparación entre variables es válida si aparecen en la misma cantidad de modelos.

# Dado que en los modelos donde están las interacciones también contienen

# a los efectos simples de los predictores, los efectos simples se encuentran en más modelos que las interacciones.

# Entonces estaría bien comparar efectos principales entre sí (línea, genero, peso\_larva, peso\_pupa) e, interacciones dobles entre sí.

# Pero no efectos principales vs. interacciones ya que las interacciones aparecen en menos modelos.

# Por ejemplo, el gráfico muestra que el peso\_larva tiene menos importancia relativa que los otros predictores simples

# **IMPORTANTE:** La utilidad del gráfico de importancia relativa está supeditada a las conclusiones de la tabla AIC obtenida con la función dredge.

# Si el modelo nulo tiene un AIC similar a los mejores modelos, el gráfico de importancia relativa pierde sentido.

```
barplot(rel_impor[5:10], main= "Importancia relativa de las variables", las=2, col=rainbow(15))
```

# Cuidado con los pesos. Si pedimos un subset o nos quedamos con los primeros, # el peso será calculado sobre ese subconjunto y no sobre el total de los modelos generados con la función dredge.

head(sel\_1)

# La salida muestra los grados de libertad del numerador (NumDF), los grados de libertad del denominador (denDF),

# El valor del estadístico F (F-value) y el valor de probabilidad asociado (p-value)

# Observar que el factor "Linea" asociado a las cajas, posee grados de libertad distintos de los demás factores asociados a las pupas.

# Llegamos a la misma conclusión que con LRT del punto 5...

# En próximos capítulos presentaremos otras opciones de inferencia para debatir.

### # 10- Tarea

# a. Plantear un modelo con peso de la pupa como pendiente aleatoria.

# b. Comparar los gráficos de ajuste con el modelo planteado anteriormente.

### # 11- Resumen

# a. Comparamos la validez de distintos modelos analizando si eran o no adecuados de acuerdo a nuestro diseño experimental (i.e.: "no.pooling", "complete.pooling" y "partial.pooling" o modelo mixto).

# b. Describimos los pasos para realizar construir un modelo de análisis: componente aleatorio, componente fijo, comprobación del modelo.

# c. Utilizamos distintos marcos de inferencia como ejemplos de alternativas para interpretar el modelo: prueba de relación de verosimilitud (LRT), inferencia multimodelo y anova.

### # 12- Agradecimiento

Datos gentilmente provistos por Artave Gabriela, Dobler Samanta, Lopez Zieher Ximena María del Laboratorio de sericultura de la Cátedra de Producciones Alternativas de la Facultad de Agronomía de la Universidad de Buenos Aires. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.

# Modelar varianzas heterogéneas

## Contenidos

<b># CAPÍTULO 5: Modelar varianzas</b> .....	<b>63</b>
<b># 1- Configuración inicial</b> .....	<b>63</b>
# 1.a- Definir el directorio de trabajo .....	63
# 1.b- Leer el archivo de datos .....	63
<b># 2- El caso: Eficiencia de consumo del ganado vacuno</b> .....	<b>63</b>
# 2.a- Pregunta de interés .....	64
# 2.b- Variable respuesta.....	64
# 2.c- Variable/s predictoras de interés.....	64
<b># 3- Identificar jerarquías</b> .....	<b>64</b>
<b># 4- Explorar los datos</b> .....	<b>64</b>
# 4.a- Outliers .....	64
# 4.b- Homogeneidad .....	65
# 4.c- Normalidad .....	65
# 4.d- Multicolinealidad .....	65
<b># 5- Construcción del modelo inicial</b> .....	<b>65</b>
# Verificación de la validez del modelo .....	65
<b># 6- Modelamos las varianzas</b> .....	<b>66</b>
# 6.a- Verificación del modelo .....	67
# 6.b- Ajustados vs. residuales del modelo VarIdent.....	68
# 6.c- Ajuste de los modelos .....	69
<b># 7- Modelos alternativos con VarIdent</b> .....	<b>70</b>
# 7.a- Verificación de los modelos .....	70
# 7.b- Ajuste y comparación de los modelos .....	71
# 7.c- Verificación del modelo .....	72
<b># 8- Modelo final</b> .....	<b>73</b>
# 8.a- Modelo final .....	73
# 8.b- Interpretación.....	73
<b># 9- Funciones de la Varianza</b> .....	<b>76</b>
<b># 10- Resumen</b> .....	<b>76</b>
# 10.a- ¿Qué hemos hecho hasta aquí? .....	76
# 10.b- ¿Qué nos queda por hacer?.....	76
<b># 11- Evaluar el componente fijo</b> .....	<b>76</b>
# Test de comparaciones múltiples.....	77
<b># 12- Gráfico descriptivo</b> .....	<b>77</b>
<b># 13- Agradecimiento</b> .....	<b>78</b>

## # CAPÍTULO 5: Modelar varianzas

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo de datos

```
# Eficiencia<-.....(Buscar ayuda y la tabla de datos "5_Eficiencia.csv")
summary(Eficiencia)
# Recordar: verificar que las variables sean leídas de manera correcta
# Corregir e indicar qué tipo de variable es cada columna
Eficiencia$Parcela<-as.factor(Eficiencia$Parcela) # sobrescribimos sobre la
misma columna
Eficiencia$fCarga<-as.factor(Eficiencia$fCarga)
summary(Eficiencia)
```

### # 2- El caso: Eficiencia de consumo del ganado vacuno

La mayor desventaja en la producción de alimentos de origen animal es su ineficiencia; en término de energía y proteína comestible por hectárea<sup>1</sup>. La creciente demanda de alimento de origen animal, ha impulsado el estudio de



factores que afectan la eficiencia biológica y económica de la producción animal. Uno de los principales factores que afectan la eficiencia biológica en la producción de carne bovina es la eficiencia de consumo entendida como la relación entre la biomasa de forraje consumida por el animal y la biomasa de forraje disponible.

Para evaluar el efecto de la carga animal (cantidad de animales por hectárea) sobre la eficiencia de consumo se

instalaron 12 parcelas en cada uno de dos sitios. Un sitio corresponde a una Pastura y el otro a un Pastizal. A las parcelas se les asignó al azar un nivel de carga animal (0,5; 1; 1,5 cab/ha. 4 repeticiones por sitio). Se determinó la eficiencia de consumo animal a lo largo de la cuatro estaciones del año. Para estimar la eficiencia de consumo se determinó la biomasa forrajera antes (Biom.pre) y después (Biom.pos) del pastoreo. El "consumo" se calculó como Biom.pre - Biom.pos y, la eficiencia (%) como el "consumo" /Biom.pre\*100. En este caso vamos a analizar los datos de otoño (2 sitios x 12 parcelas.sitio<sup>-1</sup> = 24 parcelas.estación<sup>-1</sup>). La variable respuesta es la eficiencia de consumo (%) y la variable predictora, la carga animal ambas determinadas a nivel de parcela. El sitio donde se encuentran agrupadas las parcelas generan la estructura de dependencia y será el factor aleatorio. La tabla de datos contiene las variables i) Parcela, ii) Sitio, iii) Carga y iv) Eficiencia (Efic).

<sup>1</sup><http://www.engormix.com/MA-avicultura/nutricion/articulos/produccion-animal-eficiente-t3278/141-p0.htm>



## # 2.a- Pregunta de interés

# ¿Qué efecto tiene la carga animal sobre la eficiencia de consumo?

## # 2.b- Variable respuesta

# Eficiencia de consumo (%)

## # 2.c- Variable/s predictor de interés

# Carga (3 niveles) (cab/ha): 0.5; 1; 1.5

# Estación del año (1 de 4): otoño (verano, invierno, primavera).

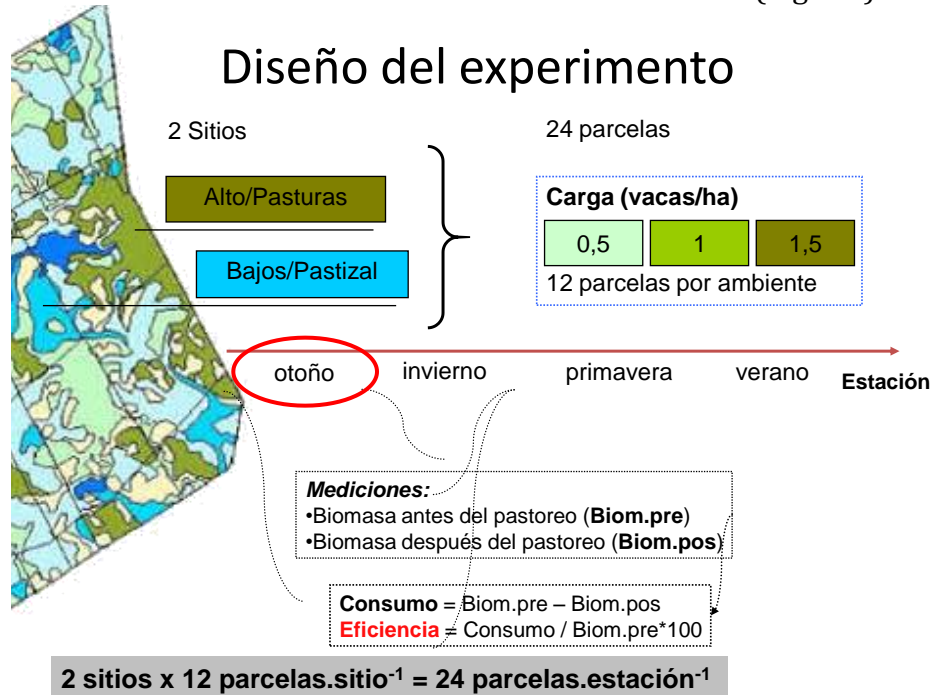
# **ATENCIÓN:** Utilizaremos datos de una sola estación (Fig. 5.1).

Figura 5. 1: diseño del experimento

## # 3- Identificar jerarquías

# Parcelas en distintos Sitios (2): Pastura vs. Pastizal

# Es decir, dos niveles: parcelas en sitios

# En este caso, la variable respuesta y las predictoras están determinadas a nivel de parcelas

## # 4- Explorar los datos

## # 4.a- Outliers

```
with(Eficiencia, dotchart(Efic, xlab="Eficiencia de consumo (%)", ylab=" Datos (# de orden)"))
```

```
with(Eficiencia, dotchart(Efic, groups=fCarga, xlab="Eficiencia de consumo (%)", ylab="Datos por nivel de carga animal"))
```

```
with(Eficiencia, dotchart(Efic, groups=Sitio, xlab="Eficiencia de consumo (%)", ylab="Datos según sitio"))
```

# No hay outliers

```
with(Eficiencia, boxplot(Efic~fCarga:Sitio, xlab="Carga animal y sitio", ylab="Eficiencia de consumo (%)"))
```



# Las varianzas no son homogéneas

# COMPLETAR LAS SENTENCIAS usando como guía el CAPÍTULO 2

#### # 4.b- Homogeneidad

#### # 4.c- Normalidad

# 4.b y c. No como diagnóstico, solo a modo exploratorio

#### # 4.d- Multicolinealidad

### # 5- Construcción del modelo inicial

# Modelo inicial de varianzas homogéneas

# Ajustamos el modelo con todos los factores

# e interacciones respetando la estructura jerárquica impuesta por el diseño.

# La carga se evaluó a nivel de parcelas y estas se encuentran "anidadas" en los sitios.

```
library("nlme")
```

```
M.lme1<- lme(Efic ~ fCarga, random=~1|Sitio, data = Eficiencia)
```

#### # Verificación de la validez del modelo

#### # 5.a- Ajustados vs. los residuales del modelo

```
plot(M.lme1)
```

# Es equivalente a

```
# E.lme1<-resid(M.lme1, type="normalized")
```

```
# F.lme1<-fitted(M.lme1)
```

```
# plot(x=F.lme1, y=E.lme1, xlab="Ajustados", ylab="Residuales estandarizados")
```

```
# abline(a=0,b=0, col="violet", lw=3)
```

```
# # Otra manera de generar este gráfico
```

```
# plot(M.lme1, which = c(1), col = 1, add.smooth = FALSE,caption = "")
```

#### # 5.b- Residuales en función de las variables

```
E.lme1<-resid(M.lme1, type="normalized")
```

layout (matrix(1:3, 1,3)) # sentencia que organiza la ventana gráfica en cuatro cuadrantes

```
boxplot(E.lme1~Eficiencia$fCarga, main="Carga", xlab="Carga animal",
ylab="Residuos normalizados de M.lme1")
```

```
boxplot(E.lme1~Eficiencia$Sitio, main="Sitio", xlab="Sitio", ylab="Residuos
normalizados de M.lme1")
```

```
boxplot(E.lme1~Eficiencia$Sitio:Eficiencia$fCarga, main="Sitio x carga",
xlab="Sitio x carga", ylab="Residuos normalizados de M.lme1")
```

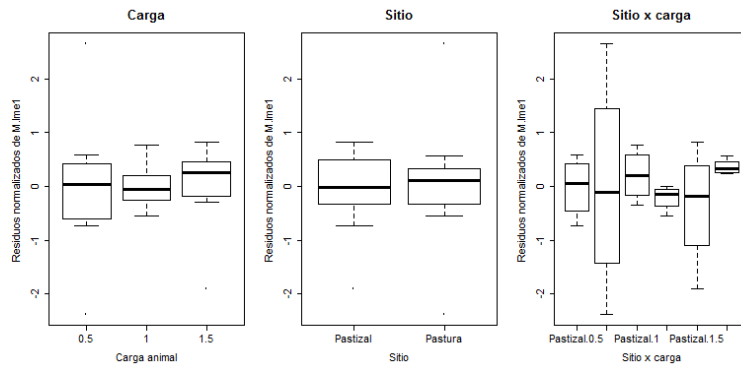


Figura 5. 2: residuales normalizados del modelo " M.lme1" en relación a los niveles de carga animal, de la calidad del sitio (Pastizal o Pastura) y, de la interacción entre ambos.

## # 6- Modelamos las varianzas

# Observamos que las varianzas son heterogéneas a nivel de Carga y a nivel de Sitio.

# Incorporar una función (varIdent) que estima una varianza diferente # para cada nivel del factor carga.

# Un aspecto interesante de esta función es que incorpora # nuevos parámetros al modelo (tantos como niveles del factor menos uno).

```
M.lme2<- lme(Efic ~ fCarga,
  random=~1|Sitio,
  weights = varIdent(form=~ 1 | fCarga),data = Eficiencia)
```

```
summary(M.lme2)
```

# Observar que aparece un nuevo término en la salida "summary(M.lme2)" # del modelo que hace referencia a la función que modela la varianza

```
Variance function:
  Structure: Different standard deviations per stratum
Formula: ~1 | fCarga
Parameter      estimates:
  0.5           1           1.5
1.0000         0.2642         0.6548
```

# El modelo estima una varianza para cada nivel de Carga (0.5, 1 y 1.5).

# Los coeficientes de la función de la varianza representan las relaciones

# entre cada varianza con respecto a una varianza de referencia.

# Esta varianza de referencia corresponde por defecto a la varianza del primer nivel.

# Por este motivo el coeficiente es 1.

# En consecuencia, esta función de la varianza agrega n-1 parámetros al modelo (n= niveles del factor)

# Finalmente, podemos estimar el desvío estándar para cada nivel de Carga

```
# Si,
```

```
Random effects:
  Formula: ~1 | Sitio
          (Intercept) Residual
StdDev:      6.899     12.19
```

```
sd_0.5= 1 * 12.19
sd_1= 0.2642 * 12.19
sd_1.5= 0.6548 * 12.19
```

```
# ó, de una manera más directa, podemos extraer sigma:
summary(M.lme2)$sigma
[1] 12.18767
```

```
# ó,
intervals(M.lme2)$sigma
      lower      est.      upper
7.214611 12.187667 20.588670
attr(,"label")
[1] "Within-group standard error:"
```

```
intervals(M.lme2)$sigma[2]
      est.
12.18767
```

```
# y los estimados de la función de la varianza.
intervals(M.lme2)$varStruct
      lower      est.      upper
1 0.1218363 0.2641894 0.5728674
1.5 0.3075499 0.6547762 1.3940236
attr(,"label")
[1] "Variance function:"
```

```
# Entonces, podemos calcular,
sd_0.5= intervals(M.lme2)$sigma[2]
# Recordemos que para el primer nivel del factor, será "1"
sd_1= intervals(M.lme2)$varStruct[1,2] * intervals(M.lme2)$sigma[2]
sd_1.5= intervals(M.lme2)$varStruct[2,2] * intervals(M.lme2)$sigma[2]
```

```
sd_0.5
      est.
12.18767
sd_1
      est.
3.219853
sd_1.5
      est.
7.980194
```

## # 6.a- Verificación del modelo

### # 6.a.1- Varianzas

```
# Verificar si se redujo la heterogeneidad de varianzas.
layout(matrix(1:2, 1,2))
plot(Eficiencia$fCarga,E.lme1, ylim=c(-3, 3), main="Modelo inicial",
cex.main=0.8, xlab="Carga animal", ylab="Residuos normalizados")
```

```
E.lme2<-resid(M.lme2,type="normalized")
plot(Eficiencia$fCarga,E.lme2, ylim=c(-3, 3), main="Modelo con VarIdent",
cex.main=0.8, xlab="Carga animal", ylab="Residuos normalizados")
```

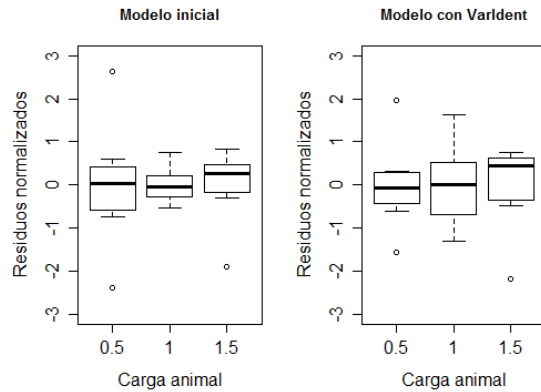


Figura 5. 3: residuales normalizados de los modelos "M.lme1" (Izq.) y "M.lme2" (Der.) en relación a los niveles de carga animal.

# Podemos comparar los desvíos estándar (sd) de los modelos sin y con modelado de las varianzas

```
tapply(E.lme1,Eficiencia$Carga,sd)
      0.5      1      1.5
1.4093560 0.4135626 0.8427193
```

```
tapply(E.lme2,Eficiencia$Carga,sd)
      0.5      1      1.5
0.9961190 0.9428701 0.9909239
```

#### # 6.a.2- Normalidad

```
layout(matrix(1:2,1,2))
qqnorm(E.lme1,ylim=c(-3,3),xlim=c(-3,3),main="Modelo inicial",
cex.main=0.8)
qqline(E.lme1)
qqnorm(E.lme2,ylim=c(-3,3),xlim=c(-3,3),main="Modelo con VarIdent",
cex.main=0.8)
qqline(E.lme2)
```

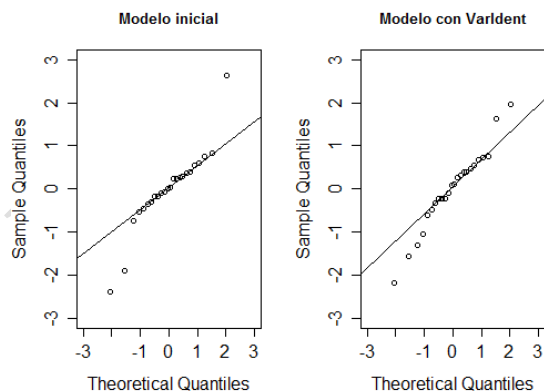


Figura 5. 4: gráficos de normalidad para los modelos "M.lme1" (Izq.) y "M.lme2" (Der.).

#### # 6.b- Ajustados vs. residuales del modelo VarIdent

```
layout(matrix(1:2,1,2))
F.lme1<-fitted(M.lme1)
plot(x=F.lme1,y=E.lme1,ylim=c(-3,3),main="Modelo inicial",cex.main=0.8,
xlab="Ajustados",ylab="Residuales normalizados")
```

```
abline(a=0,b=0, col="violet", lw=3)
```

```
F.lme2<-fitted(M.lme2)
plot(x=F.lme2, y=E.lme2, ylim=c(-3, 3), main="Modelo con VarIdent",
cex.main=0.8, xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

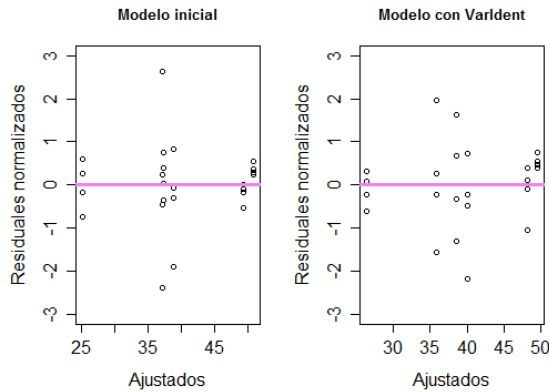


Figura 5. 5: residuales normalizados en relación a los valores ajustados para los modelos "M.lme1" (izq.) y "M.lme2" (der.).

```
# Alternativas:
# plot(M.lme2, which = c(1), col = 1, add.smooth = FALSE,caption = "")
# plot(M.lme2)
```

### # 6.c- Ajuste de los modelos

#### # 6.c.1- Test del cociente de verosimilitud: LRT

```
anova(M.lme1,M.lme2)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
M.lme1	1	5	168.6092	173.8318	-79.30459			
M.lme2	2	7	163.4506	170.7623	-74.72530	1 vs 2	9.158575	0.0103

```
# ¿Cómo se estima el L.Ratio?
L.Ratio = - 2 * (-79.30459 - (-74.72530))
L.Ratio
[1] 9.15858
```

```
# En este caso los grados de libertad son 2, la diferencia en el número de
parámetros (En este caso, 7-5)
# Podemos estimar la probabilidad de encontrar un valor de L.Ratio => 9.159
# Bajo la hipótesis nula que indica que no hay diferencias en la verosimilitud de
los modelos fuese cierta.
```

```
1 - pchisq(L.Ratio,2)
[1] 9.15858
```

```
# Observar que es el mismo valor obtenido a partir de la función
anova(M.lme1,M.lme2)
```

#### # 6.c.2- Criterio de información de Akaike (AIC)

```
# ¿Cómo se calcula el AIC?
```

# Modelo inicial: sin heterogeneidad de varianzas

AIC(M.lme1)  
[1] 168.6092

$-2 * -79.30459 + 2 * 5$   
[1] 168.6092

# Son 5 parámetros = 1 ordenada, 2 para completar los niveles de Carga, 1 factor aleatorio Sitio y 1 varianza.

# Recordemos que en la función de verosimilitud utilizando la distribución normal utilizamos

# tanto la media como la varianza.

# Modelo con la función VarIdent

AIC(M.lme2)  
[1] 163.4506

$-2 * -74.72530 + 2 * 7$   
[1] 163.4506

# Son 7 parámetros porque agregamos a los anteriores 2 parámetros con la función de la varianza

# El modelo M.lme2 es sensiblemente mejor que el modelo M.lme1.

# Sin embargo, aun el gráfico de los residuales vs. los ajustados muestra varianzas heterogéneas.

# Podemos generar nuevos modelos con funciones de la varianza para Sitio y/o la interacción Carga\*Sitio.

### # 7- Modelos alternativos con VarIdent

```
M.lme3<- lme(Efic ~ fCarga,
  random=~1|Sitio,
  weights = varIdent(form=~ 1| Sitio),data = Eficiencia)
```

```
M.lme4<- lme(Efic ~ fCarga,
  random=~1|Sitio,
  weights = varIdent(form=~ 1 | fCarga*Sitio),data = Eficiencia)
```

### # 7.a- Verificación de los modelos

#### # 7.a.1- Varianzas

# Residuales vs. predictores

```
E.lme3=resid(M.lme3,type="normalized")
plot(Eficiencia$Sitio,E.lme3, ylim=c(-3, 3), main="Modelo con VarIdent(Sitio)",
  cex.main=0.8, cex.axis=0.8, xlab="Carga", ylab="Residuales normalizados")
```

```
E.lme4=resid(M.lme4,type="normalized")
Eficiencia$inter<-interaction(Eficiencia$fCarga,Eficiencia$Sitio, sep="_")
plot(Eficiencia$inter,E.lme4, las=3, ylim=c(-3, 3), main="Modelo con
  VarIdent(Carga*Sitio)", cex.main=0.8, cex.axis=0.6, xlab="", ylab="Residuales
  normalizados")
```

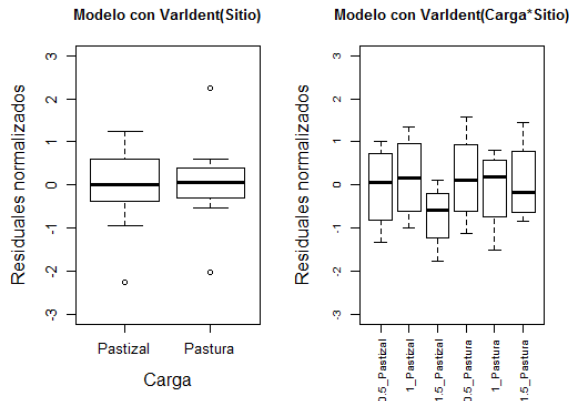


Figura 5. 6: residuales normalizados en relación a la calidad del sitio (Pastizal o Pastura. Izq.) y, la interacción entre nivel de carga x calidad de sitio para los modelos "M.lme3" y "M.lme4", respectivamente. Dertalles de los modelos están descriptos en el cuerpo del texto.

# Residuales vs. ajustados

```
plot(M.lme3, which = c(1), col = 1, add.smooth = FALSE,caption = "")
plot(M.lme4, which = c(1), col = 1, add.smooth = FALSE,caption = "")
```

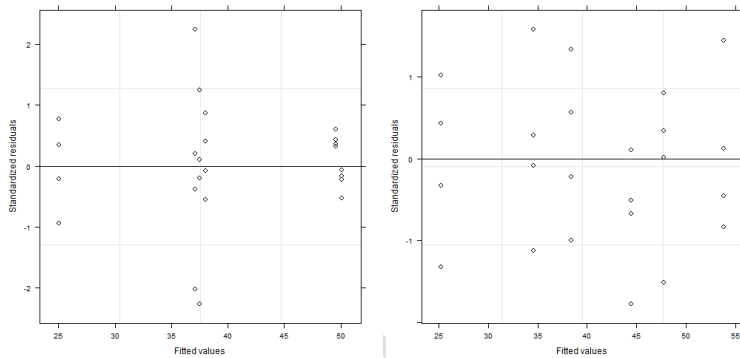


Figura 5. 7: residuales normalizados en relación a los valores ajustados para los modelos "M.lme3" (Izq.) y "M.lme4" (Der.). Dertalles de los modelos están descriptos en el cuerpo del texto.

```
tapply(E.lme3,Eficiencia$Sitio,sd)
tapply(E.lme4,Eficiencia$inter,sd)
```

# **REALICE** las sentencias que completan la verificación del modelo

# 7.a.2- Normalidad

# 7.a.3- Ajustados vs. residuales

# 7.b- Ajuste y comparación de los modelos

```
anova(M.lme2,M.lme3, M.lme4)
```

# Para comparar entre varios modelos, otra opción es usar AIC en lugar del LRT.

```
library("bbmle")
```

```
AICtab(M.lme2,M.lme3, M.lme4, weights = T, delta = TRUE, base=T, sort = TRUE)
```

	AIC	dAIC	df	weight
M.lme4	155.5	0.0	10	0.9802
M.lme2	163.5	7.9	7	0.0186
M.lme3	169.0	13.4	6	0.0012

# Investigar en la ayuda esta función

?AICtab

```
# Observar el AIC, el delta y el peso para comparar el ajuste de los modelos
# Optamos por el modelo "M.lme4"
```

### # 7.c- Verificación del modelo

```
layout(matrix(1:4, 2,2))
```

#### # 7.c.1- Residuales vs ajustados

```
F.lme4<-fitted(M.lme4)
plot(x=F.lme4, y=E.lme4, ylim=c(-3, 3), main="Modelo con
VarIdent(Carga*Sitio)", cex.main=0.8, xlab="Ajustados", ylab="Residuales
normalizados")
abline(0,0,col="red",lwd=3)
```

#### # 7.c.2- Residuales vs las variables

```
boxplot(E.lme4~Eficiencia$fCarga, ylim=c(-3, 3), main="Carga", xlab="Carga
animal", ylab="Residuales normalizados")
boxplot(E.lme4~Eficiencia$Sitio, ylim=c(-3, 3), main="Sitio", xlab="Sitio",
ylab="Residuales normalizados")
# Predichos vs observados
plot(x=Eficiencia$Efic, y=F.lme4, xlab="Observados", ylab="Predichos")
abline(0,1,col="red",lwd=3)
```

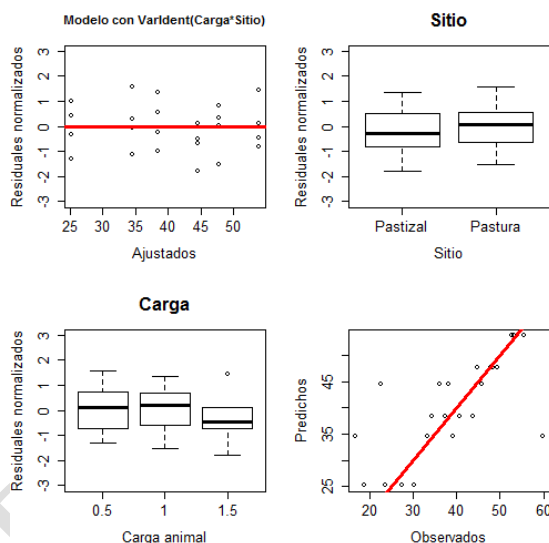


Figura 5. 8: residuales normalizados del modelo "M.lme4" en relación a los valores ajustados por el modelo, la calidad del sitio y, el nivel de carga. Valores ajustados por el modelo en relación a los observados en el cuadrante inferior derecho.

#### # 7.c.3- Normalidad

```
layout(matrix(1:2,1,2))
qqnorm(E.lme4, main="Modelo con VarIdent(Carga*Sitio)", cex.main=0.8)
qqline(E.lme4)
# Comparar con el modelo inicial
qqnorm(E.lme1, main="Modelo inicial", cex.main=0.8)
qqline(E.lme1)
```



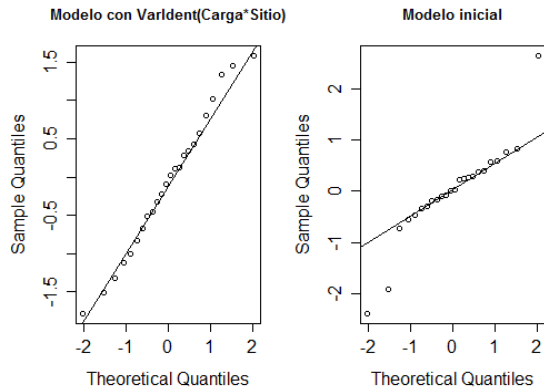


Figura 5. 9: gráficos de normalidad para los modelos "M.lme4" (izq.) y "M.lme1" (der.).

### # 8- Modelo final

#### # 8.a- Modelo final

M.lme4\$call

```
# lme.formula(fixed = Efic ~ fCarga, data = Eficiencia, random = ~1 |
# Sitio, weights = varIdent(form = ~1 | fCarga * Sitio))
```

# El modelo se puede representar de manera matemática como:

$$Y_i \sim N(\mu_i, \sigma_{\epsilon}^2 \text{ carga}_i * \text{sitio}_i)$$

$$\mu_i = \beta_{0j[i]} + \beta_1 * \text{carga}_{1i} + \beta_2 * \text{carga}_{1,5i}$$

$$\beta_{0j} \sim N(\mu_{\beta_0}, \sigma_{\beta_0}^2)$$

$i = 1, 2, 3, \dots, 24$  parcelas  
 $j = 1, 2$  sitios

**Donde,**

$Y_i$  = eficiencia de consumo

Coefficientes:

$\beta_{0j}$  = intercepción

$\beta_1$  = carga 1

$\beta_2$  = carga 1,5

$\sigma_{\beta_0}^2$  = varianza entre sitios

$\sigma_{\epsilon}^2$  = varianzas residuales

#### # 8.b- Interpretación

summary(M.lme4)

```
Linear mixed-effects model fit by REML
Data: Eficiencia
AIC    BIC    logLik
155.5  166    -67.76
```

# Practicar la fórmula de AIC

**# AIC:**

```
-2*summary(M.lme4)$logLik + 2 * 10
# en R log es el logaritmo natural
# el modelo consta de 10 parámetros:
# - tres niveles de Carga
# - uno para el factor aleatorio Sitio
# - cinco parámetros de la función de la varianza (ver debajo)
# - una para la varianza
```

```
Random effects:
  Formula: ~1 | Sitio
          (Intercept)  Residual
StdDev:      6.756      4.836
```

#  $6.756^2$  representa la variación entre sitios o la co-varianza entre parcelas dentro del mismo sitio.  
 #  $\sigma^2 = 4.836^2$  es la variación residual entre parcelas dentro de un sitio.  
 #  $d^2 + \sigma^2 = 6.756^2 + 4.836^2$ , es la variación total no explicada entre parcelas.  
 #  $\rho = d^2 / (d^2 + \sigma^2) = 6.75^2 / (6.756^2 + 4.836^2)$  es la correlación entre parcelas dentro de un sitio (intraclass correlation)

```
Variance function:
  Structure: Different standard deviations per stratum
  Formula: ~1 | fCarga * Sitio
  Parameter estimates:
0.5*Pastizal  1*Pastizal  1.5*Pastizal  0.5*Pastura  1*Pastura
1.0000      0.8366    2.5528      3.2830      0.4177
1.5*Pastura
0.2545
```

# **Recordar:** estos valores son el "factor de multiplicación"  
 # y muestran la relación de cada grupo de datos con el desvío residual  
 # en este caso el desvío sería:

```
M.lme4$sigma
[1] 4.836338
# y la varianza residual (sigma):
M.lme4$sigma^2
[1] 23.39017
```

# **ATENCIÓN:** los "factores de multiplicación" hay que multiplicarlos por el desvío (4.836) y NO por la varianza.  
 # A este 4.836 es al que en la salida del gls se llama incorrectamente "error" estándar.

```
Fixed effects: Efic ~ fCarga
Value Std.Error DF t-value p-value
(Intercept) 29.92 5.373 20 5.568 0e+00
fCarga1 13.16 2.871 20 4.586 2e-04
fCarga1.5 19.21 3.007 20 6.390 0e+00
Correlation:
  (Intr) fCarg1
fCarga1 -0.425
fCarga1.5 -0.425 0.924
```

```
Standardized within-Group Residuals:
  Min       Q1       Med       Q3       Max
-1.77941 -0.71143 -0.03344  0.46525  1.58507
```

```
Number of Observations: 24
Number of Groups: 2
```

# A los modelos podemos pedirle varias cosas....

```
names(summary(M.lme4))
```

```
# Por ejemplo,
```

```
summary(M.lme4)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	29.91847	5.373000	20	5.568300	1.888845e-05
fCarga1	13.16318	2.870580	20	4.585549	1.792692e-04
fCarga1.5	19.21524	3.007214	20	6.389716	3.108686e-06

```
names(M.lme4)
```

```
# Por ejemplo,
```

```
M.lme4$coefficients
```

```
$fixed
```

	fCarga1	fCarga1.5
(Intercept)	29.91847	19.21524

```
$random
```

```
$random$Sitio
```

	(Intercept)
Pastizal	-4.66997
Pastura	4.66997

# Los intervalos de confianza de los valores estimados que observamos en el punto anterior

```
intervals(M.lme4)
```

```
Approximate 95% confidence intervals
```

```
Fixed effects:
```

	lower	est.	upper
(Intercept)	18.710592	29.91847	41.12635
fCarga1	7.175259	13.16318	19.15111
fCarga1.5	12.942303	19.21524	25.48818

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

```
Random Effects:
```

Level: Sitio	lower	est.	upper
sd((Intercept))	1.561737	6.755823	29.2246

```
Variance function:
```

	lower	est.	upper
1*Pastizal	0.27451886	0.8365750	2.5493977
1.5*Pastizal	0.86792158	2.5527810	7.5083870
0.5*Pastura	1.15862790	3.2830423	9.3026991
1*Pastura	0.13666402	0.4177354	1.2768752
1.5*Pastura	0.08290366	0.2545187	0.7813864

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

```
within-group standard error:
```

	lower	est.	upper
	2.209749	4.836338	10.584990

### # 9- Funciones de la Varianza

# Es interesante conocer las funciones disponibles para modelar la varianza  
?varClasses

# Con variables continuas se pueden utilizar las siguientes funciones de la varianza:

```
# vf.vf <- varFixed(~continua)
```

```
# vf.vp <- varPower(form=~continua)
```

```
# vf.vp2 <- varPower(form=~Bloque|continua)
```

```
# vf.ve <- varExp(form=~continua)
```

```
# vf.ve2 <- varExp(form=~Bloque|continua)
```

# Existe una función que permite combinar las anteriores. Por ejemplo,

```
# vf.vc<-varComb(varIdent(form=~1|Bloque),varPower(form=~continua))
```

# Recordar que podemos incorporar cualquier función con un poco más de notación.

### # 10- Resumen

#### # 10.a- ¿Qué hemos hecho hasta aquí?

# - Escribimos nuestro modelo de efectos mixtos (o de "partial pooling").

# que contempla la estructura dada por el diseño del experimento y el método de muestreo.

# - Encontramos la estructura óptima para los componentes aleatorios

# (mediante funciones de la varianza).

#### # 10.b- ¿Qué nos queda por hacer?

# - Evaluar el componente fijo del modelo (método ML).

# - Presentar el modelo final utilizando el método REML de estimación.

### # 11- Evaluar el componente fijo

# Para esto utilizar la función "update"

```
M.lme4ML<-update(M.lme4, method="ML")
```

```
summary(M.lme4ML)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	29.64679	4.187145	20	7.080432	7.284977e-07
fCarga1	13.60147	2.854551	20	4.764837	1.182487e-04
fCarga1.5	19.77705	2.989144	20	6.616292	1.918021e-06

# En este caso, hay una sola variable predictora

# por lo tanto, podemos comparar el modelo con el modelo nulo

```
M.lme4ML2<-update(M.lme4ML, ~.-fCarga)
```

```
anova(M.lme4ML, M.lme4ML2)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
M.lme4ML	1	10	165.98	177.76	-72.99			
M.lme4ML2	2	8	182.98	192.41	-83.49	1 vs 2	21.0001	<.0001

# Observar el resultado del Test de relaciones de máxima verosimilitud.

# Hay diferencias en el ajuste entre los modelos.

# Además, el modelo que tiene la carga como factor (M.lme4ML)

# tiene mejor ajuste (menor AIC) por lo tanto, el modelo debería incluir a este predictor.

```
# Ahora bien, hay tres niveles de carga,
# ¿Hay diferencia entre los niveles de la variable Carga?
```

### # Test de comparaciones múltiples

```
library(multcomp)
summary(glht(M.lme4, linfct=mcp(fCarga="Tukey")))
```

Simultaneous Tests for General Linear Hypotheses  
Multiple Comparisons of Means: Tukey Contrasts

```
Fit: lme.formula(fixed = Efic ~ fCarga, data = Eficiencia, random =
~1 |
  Sitio, weights = varIdent(form = ~1 | fCarga * Sitio))
```

Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z )
1 - 0.5 == 0	13.16	2.87	4.59	1.2e-05 ***
1.5 - 0.5 == 0	19.22	3.01	6.39	< 1e-06 ***
1.5 - 1 == 0	6.05	1.16	5.23	< 1e-06 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
(Adjusted p values reported -- single-step method)

### # 12- Gráfico descriptivo

```
layout(matrix(1:1,1,1))
plot(x=Eficiencia$Carga, y = Eficiencia$Efic,
     type = "p",lwd=2 ,col="gray", xlim=c(0,2), ylim=c(0,65),
     ylab=" Eficiencia en el consumo (%)", xlab="Carga animal")

# Agregamos puntos que indican los estimados del modelo general
points(0.5, fixef(M.lme4)[1], cex=2,pch=18, col="blue", lwd=3)
points(1, fixef(M.lme4)[1]+fixef(M.lme4)[2], cex=2,pch=18, col="blue", lwd=3)
points(1.5, fixef(M.lme4)[1]+fixef(M.lme4)[3], cex=2,pch=18, col="blue", lwd=3)

# Valores estimados para el sitio de Pastizal
points(0.5, ranef(M.lme4)[1,1]+fixef(M.lme4)[1], cex=2,pch=18, col="red",
lwd=3)
points(1, ranef(M.lme4)[1,1]+fixef(M.lme4)[1]+fixef(M.lme4)[2], cex=2,pch=18,
col="red", lwd=3)
points(1.5, ranef(M.lme4)[1,1]+fixef(M.lme4)[1]+fixef(M.lme4)[3],
cex=2,pch=18, col="red", lwd=3)

# Valores estimados para el sitio de Pastura
points(0.5, ranef(M.lme4)[2,1]+fixef(M.lme4)[1], cex=2,pch=18, col="green",
lwd=3)
points(1, ranef(M.lme4)[2,1]+fixef(M.lme4)[1]+fixef(M.lme4)[2], cex=2,pch=18,
col="green", lwd=3)
points(1.5, ranef(M.lme4)[2,1]+fixef(M.lme4)[1]+fixef(M.lme4)[3],
cex=2,pch=18, col="green", lwd=3)

legend("bottomright", c("General", "Pastizal", "Pastura"), col=c("blue", "red",
"green"), pch=16, bty="o",bg="light grey" )
```

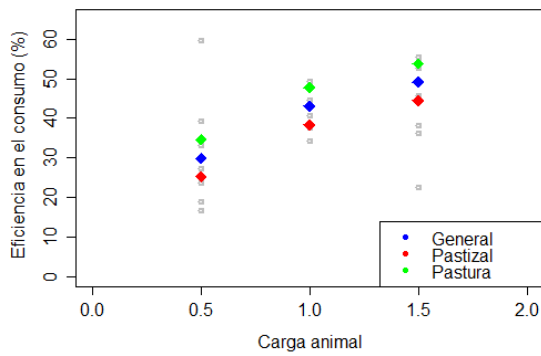


Figura 5. 10: eficiencia de consumo en función del nivel de carga animal. Los puntos azules muestran los valores ajustados por el modelo general. Los puntos rojo y verde para los sitios (aleatorios) de Pastizal y Pastura, respectivamente.

### # 13- Agradecimiento

Datos gentilmente provistos por Gonzalo Irisarri y Martín Oesterheld de la Facultad de Agronomía de la Universidad de Buenos Aires. Irisarri, G. N. 2012. Variación espacial y temporal de la producción primaria neta aérea y secundaria. Tesis Doctoral, EPG-FAUBA. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.

# Modelar varianzas heterogéneas II

## Contenidos

<b># CAPÍTULO 6: Modelar las varianzas II .....</b>	<b>80</b>
<b># 1- Configuración inicial.....</b>	<b>80</b>
# 1.a- Definir el directorio de trabajo .....	80
# 1.b- Leer el archivo de datos .....	80
<b># 2- El caso: Determinantes de la calidad maltera de cebada cervecera.....</b>	<b>80</b>
# 2.a- Pregunta de interés .....	81
# 2.b- Variable respuesta.....	81
# 2.c- Variable/s predictoras de interés .....	81
<b># 3- Identificar jerarquías .....</b>	<b>81</b>
<b># 4- Explorar los datos .....</b>	<b>82</b>
# 4.a- Outliers .....	82
# 4.b- Homogeneidad .....	83
# 4.c- Normalidad .....	83
# 4.d- Multicolinealidad .....	83
<b># 5- Construcción del modelo inicial .....</b>	<b>83</b>
# Verificación de la validez del modelo .....	83
<b># 6- Modelamos las varianzas .....</b>	<b>84</b>
# 6.a- Verificación del modelo .....	84
# 6.b- Ajustados vs. residuales del modelo VarIdent.....	85
# 6.c- Ajuste de los modelos .....	85
<b># 7- Modelos alternativos con VarIdent .....</b>	<b>85</b>
# 7.a- Modelamos las varianzas para los grupos N:S .....	85
# 7.b- Modelamos las varianzas a nivel de PMS_gr.....	87
<b># 8- Ajustar el componente fijo.....</b>	<b>92</b>
# 8.a- ANOVA .....	93
# 8.b- Test de cociente de verosimilitud (LRT) .....	93
# 8.c - Función Anova .....	95
<b># 9- Modelo final .....</b>	<b>95</b>
# 9.a- Presentación del modelo.....	95
# 9.b- Gráficos de ajuste del modelo .....	96
<b># 10- Graficar el modelo promedio general.....</b>	<b>97</b>
<b># 11- Graficar los efectos aleatorios .....</b>	<b>99</b>
<b># 12- Respuesta a la pregunta .....</b>	<b>107</b>
<b># 13- Agradecimiento.....</b>	<b>107</b>

## # CAPÍTULO 6: Modelar las varianzas II

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo de datos

```
# horde<-.....(Buscar ayuda y la tabla de datos "6_Fertiliza.csv")
summary(horde)
```

# Tenemos una tabla de datos con distinto número de parcelas por localidad

# dado que no se hicieron determinaciones de hordeínas en todos los casos.

# En consecuencia, también quedaron desbalanceados los tratamientos con o sin fertilización con nitrógeno y con azufre.

### # 2- El caso: Determinantes de la calidad maltera de cebada cervecera

Efecto de la fertilización sobre la producción de hordeínas en cebada



Las hordeínas son proteínas características del grano de cebada que contienen nitrógeno y azufre. El grano de cebada produce hordeínas B, C, D y gamma. La cantidad de estas hordeínas en el grano influye sobre la calidad maltera para producción de cerveza. Sin embargo, aun se sabe poco respecto de la producción de cada una de estas hordeínas a lo largo de la ontogenia (o formación) del grano, el impacto de la fertilización y, su relación con la calidad maltera de la cebada. Para estudiar estos aspectos se realizó un experimento bajo condiciones ambientales naturales. En 4 localidades de la región pampeana (Junin, Baigorrita, Tiburcio y Campito) se instalaron 16 parcelas (Total, 64 parcelas). En cada localidad se aplicaron tratamientos de fertilización según el arreglo factorial de los factores fertilización con nitrógeno y fertilización con azufre (2 niveles cada uno. En total 4 repeticiones de los 4 tratamientos por localidad). A lo largo de la ontogenia del grano, es decir, en fechas sucesivas después de antesis se tomaron muestras de 10 espigas de cebada por parcela. Las muestras fueron utilizadas para determinar el número de granos, el peso de mil semillas (g), la cantidad de hordeínas B, C, D y Gamma (Las determinaciones de hordeínas se realizaron mediante HPLC y el valor en la tabla de datos corresponde al área determinada por el equipo).

Este experimento lo utilizaremos para evaluar: i) el efecto de la fertilización sobre la producción de hordeínas B y, ii) el efecto de la fertilización sobre la producción de hordeínas B durante la ontogenia del grano.

i) Para evaluar la influencia del nitrógeno y el azufre en la producción de hordeínas B en grano de cebada utilizaremos los valores de hordeínas B a los 36 días después de antesis. En este momento se considera que el grano ha alcanzado el máximo nivel de hordeínas. Las variables predictoras serán: a) la fertilización con nitrógeno (N, dos niveles), b) la fertilización con azufre (S, dos niveles) y, c) el peso de mil semillas (g). La estructura de dependencia está dada por las parcelas en los ambientes.



- ii) Para evaluar el efecto de la fertilización a los largo de la ontogenia del grano, agregamos el tiempo (días post antesis) a las variables predictoras anteriores. La variable tiempo contempla de seis mediciones por parcela (Fig. 6.1).

### # Diseño del experimento

#### Diseño del experimento

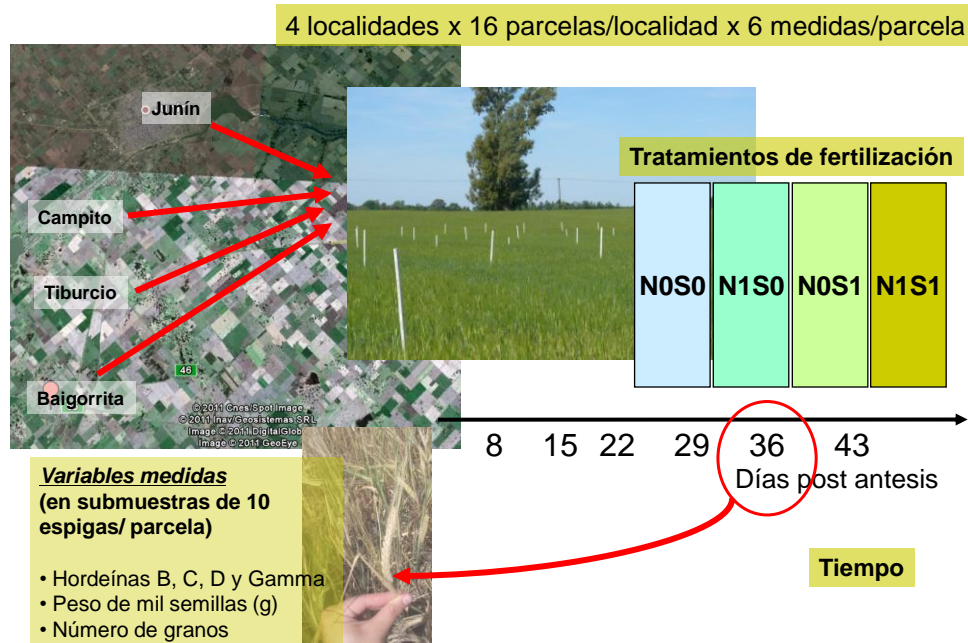


Figura 6. 1: esquema del diseño experimental. En este capítulo utilizaremos solo los datos correspondientes a los 36 días post antesis.

### # 2.a- Pregunta de interés

- # ¿Qué influencia tienen la disponibilidad de Nitrógeno y de Azufre en el suelo sobre la producción de hordeínas B en grano de cebada?
- # Las hordeínas son proteínas características del grano de cebada que contienen nitrógeno y azufre en su estructura molecular.
- # El grano de cebada produce distintos tipos de hordeínas: B, C, D y gamma.
- # La cantidad de estas hordeínas en el grano influye sobre la calidad maltera para producción de cerveza.

### # 2.b- Variable respuesta

- # Contenido de hordeínas B (Tomamos un tipo de hordeínas como ejemplo)
- # La cuantificación de hordeínas no posee unidades. Es el área bajo la curva que determina el equipo (HPLC) utilizado para hacer la determinación.

### # 2.c- Variable/s predictoras de interés

- # Fertilización: Factorial 2 Factores (N y S) con 2 niveles cada uno
- # Peso de mil semillas (granos, en gramos)

### # 3- Identificar jerarquías

- # ¿Cuáles son las escalas y qué variables se midieron en cada una?
- # Parcelas (16 por Localidad): Tratamientos de Fertilización
- # Peso de mil semillas (granos)
- # Cuantificación de hordeínas a partir de una submuestra de "n" espigas por parcela a los 35 (+-1) días post antesis.

```
# Localidades [4: J (Junin), B (Baogorrita), T (Tiburcio), C (Campito)]
```

#### # 4- Explorar los datos

```
# NO diagnóstico, solo a modo exploratorio
```

##### # 4.a- Outliers

###### # 4.a.1- Gráficos de cajas

```
# Trabajaremos con Hord_B
with(horde, plot(Localidad, Hord_B, xlab="Localidad", ylab="Hordeina B"))
with(horde, plot(N, Hord_B, xlab="Fertilización con N", ylab="Hordeina B"))
with(horde, plot(S, Hord_B, xlab="Fertilización con S", ylab="Hordeina B"))
with(horde, plot(N:S, Hord_B, xlab="Fertilización", ylab="Hordeina B", las=3))
with(horde, plot(PMS_gr, Hord_B, xlab="Peso de mil semillas (g)",
ylab="Hordeina B"))
```

```
# Idem
```

```
# with(horde, boxplot(Hord_B~Localidad, ylab="Hordeina B",
xlab="Localidad"))
# with(horde, boxplot(Hord_B~N, ylab="Hordeina B", xlab="Fertilización con
N"))
# with(horde, boxplot(Hord_B~S, ylab="Hordeina B", xlab="Fertilización con S"))
# with(horde, boxplot(Hord_B~Localidad*N*S, ylab="Hordeina B", las=3))
```

###### # 4.a.2- Gráficos distribución de puntos

```
library("lattice")
with(horde, xyplot(Hord_B ~PMS_gr|Localidad, xlab="Peso de mil semillas (g)",
ylab="Hordeinas"))
# Observar que la distribución de los pesos no fue similar entre localidades.
# Tampoco la cuantificación de hordeínas.
with(horde, xyplot(Hord_B ~PMS_gr|N, xlab="Peso de mil semillas (g)",
ylab="Hordeinas"))
with(horde, xyplot(Hord_B ~PMS_gr|S, xlab="Peso de mil semillas (g)",
ylab="Hordeinas"))
with(horde, xyplot(Hord_B ~PMS_gr|S*N, xlab="Peso de mil semillas (g)",
ylab="Hordeinas"))
# Observar la presencia de relaciones entre las variables.
```

###### # 4.a.3- Gráficos distribución de puntos II

```
with(horde, dotchart(Hord_B, ylab="Conjunto de datos", xlab="Hordeínas"))
with(horde, dotchart(Hord_B, groups=Localidad, ylab="Datos por Localidad",
xlab="Hordeínas"))
# Observar diferencias en los valores de hordeínas entre localidades
# Este tipo de gráfico también ayuda a observar diferencias en la variabilidad de
los datos entre localidades.
with(horde, dotchart(Hord_B, groups=N, ylab="Datos según fertilización",
xlab="Hordeinas"))
with(horde, dotchart(Hord_B, groups=S, ylab="Datos según fertilización",
xlab="Hordeinas"))
```

# ¿Observamos la presencia de outliers en este caso?

# COMPLETAR LAS SENTENCIAS usando como guía el CAPÍTULO 2

#### # 4.b- Homogeneidad

#### # 4.c- Normalidad

#### # 4.d- Multicolinealidad

# Puede existir colinealidad múltiple es decir, entre una variable predictora y las demás de manera conjunta.

# Una forma de evaluar este tipo de colinealidad es comparar el  $R^2$  de funciones lineales donde cada una de las predictoras funciona como variable respuesta.

```
col1<-lm(PMS_gr~N*S, data=horde)
cor(horde$PMS_gr, fitted(col1))^2
[1] 0.02476411
```

# En este caso, PMS\_gr es la única variable cuantitativa.

# Si hubiera otras variables cuantitativas se realizan todos los modelos posibles.

# La variable predictora que genera un alto  $R^2$  presenta colinealidad con las demás variables predictoras del modelo y debería ser excluida.

# En este caso, el  $R^2$  de la relación es bajo por lo tanto, la incluimos en el modelo.

#### # 5- Construcción del modelo inicial

# Incluir todos los factores e interacciones.

```
library("nlme")
mod<-lme(Hord_B~N*S*PMS_gr, random=~1|Localidad, data=horde)
```

#### # Verificación de la validez del modelo

##### # 5.a- Ajustados vs. los residuales del modelo

```
Er<-resid(mod, type="normalized")
Fit<-fitted(mod)
plot(x=Fit, y=Er, xlab="Ajustados", ylab="Residuales estandarizados")
abline(a=0, b=0, col="red", lw=3)
```

##### # 5.b- Residuales en función de las variables predictoras

```
boxplot(Er~horde$Localidad, main="Localidad", xlab="Localidad",
ylab="Residuales estandarizados")
# Observar heterogeneidad entre las varianzas
boxplot(Er~horde$N, main="N", ylab="Residuales estandarizados")
boxplot(Er~horde$S, main="S", ylab="Residuales estandarizados")
boxplot(Er~horde$S:horde$N, main="S", ylab="Residuales estandarizados",
las=3)
plot(Er~horde$PMS_gr, main="Peso mil semillas", ylab="Residuales
estandarizados")
abline(a=0, b=0, col="red", lw=3)
```

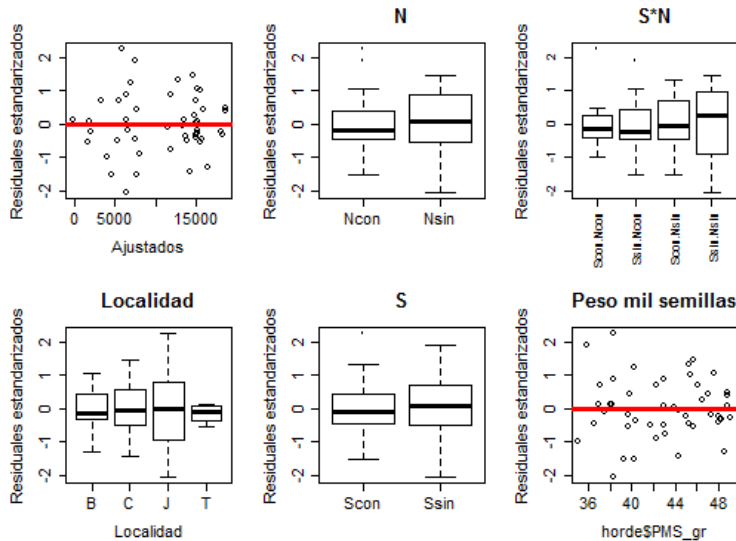


Figura 6. 2: residuales normalizados del modelo " mod " en relación a los valores ajustados por el modelo, la Localidad, en nivel de fertilización con nitrógeno (N), el nivel de fertilización con azufre (S), la interacción N x S y, el peso de mil semillas.

```
tapply(Er,horde$Localidad,sd)
tapply(Er,horde$N,sd)
tapply(Er,horde$S,sd)
tapply(Er,horde$S:horde$N,sd)
```

## # 6- Modelamos las varianzas

```
# Parece haber problemas de heterogeneidad a nivel de Localidad y, de la
# interacción N:S
# Dado que son factores, incorporamos la función (varIdent) que pondera una
# varianza diferente para cada nivel del factor.
# Recordamos que esta función incorpora nuevos parámetros al modelo (tantos
# como niveles del factor-1)
mod2<-lme(Horde_B~N*S*PMS_gr, random=~1|Localidad,
          weights = varIdent(form=~ 1 | Localidad), data=horde)
```

### # 6.a- Verificación del modelo

#### # 6.a.1- Varianzas

```
# Verificamos si se redujo la heterogeneidad de varianzas.
layout(matrix(1:2, 1,2))
boxplot(Er~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo inicial",
        cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")

Er2<-resid(mod2,type="normalized")
boxplot(Er2~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
        cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")
# Si los gráficos no se observan bien dentro de la ventana de RStudio pueden
# llevarlos a una ventana externa haciendo click en el botón "zoom"
```

```
tapply(Er,horde$Localidad,sd)
```

```
tapply(Er2,horde$Localidad,sd)
```

### # 6.a.2- Normalidad

```
layout(matrix(1:2,1,2))
qqnorm(Er, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo inicial",
cex.main=0.8)
qqline(Er)
qqnorm(Er2, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo con VarIdent",
cex.main=0.8)
qqline(Er2)
```

### # 6.b- Ajustados vs. residuales del modelo VarIdent

```
layout(matrix(1:2, 1,2))
plot(x=Fit, y=Er, ylim=c(-2.5, 2.5), main="Modelo inicial", cex.main=0.8,
xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

```
Fit2<-fitted(mod2)
plot(x=Fit2, y=Er2, ylim=c(-2.5, 2.5), main="Modelo con VarIdent", cex.main=0.8,
xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
# El patrón de distribución observado en el modelo inicial no aparece en este
modelo.
```

### # 6.c- Ajuste de los modelos

#### # 6.c.1- Test del cociente de verosimilitud: LRT

```
anova(mod, mod2)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
mod	1	10	814.4985	831.3873	-397.2493			
mod2	2	13	810.4730	832.4285	-392.2365	1 vs 2	10.02548	0.0184

```
# Observamos el resultado del LRT. La función de la varianza mejoró
# el ajuste respecto del modelo inicial.
```

### # 7- Modelos alternativos con VarIdent

#### # 7.a- Modelamos las varianzas para los grupos N:S

```
mod3<-lme(Hord_B~N*S*PMS_gr, random=~1|Localidad,
weights = varIdent(form=~ 1 | N*S), data=horde)
```

#### # 7.a.1- Verificación de los modelos

```
# - Varianzas
# Residuales vs. predictores
# Localidad
layout(matrix(1:3, 1,3))
boxplot(Er~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo inicial",
cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")
boxplot(Er2~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")
```

```
Er3<-resid(mod3, type="normalized")
```

```
boxplot(Er3~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo VarIdent(N*S)",
cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")
```

```
tapply(Er,horde$Localidad,sd)
tapply(Er2,horde$Localidad,sd)
tapply(Er3,horde$Localidad,sd)
```

```
# Nitrogeno: Azufre
```

```
layout(matrix(1:3, 1,3))
boxplot(Er~horde$N:horde$$, ylim=c(-2.5, 2.5), main="Modelo inicial",
cex.main=0.8, xlab="N*S", ylab="Residuales estandarizados")
boxplot(Er2~horde$N:horde$$, ylim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
cex.main=0.8, xlab="N*S", ylab="Residuales estandarizados")
boxplot(Er3~horde$N:horde$$, ylim=c(-2.5, 2.5), main="Modelo VarIdent(N*S)",
cex.main=0.8, xlab="N*S", ylab="Residuales estandarizados")
```

```
tapply(Er,horde$N:horde$$,sd)
tapply(Er2,horde$N:horde$$,sd)
tapply(Er3,horde$N:horde$$,sd)
```

```
# - Normalidad
```

```
layout(matrix(1:3,1,3))
qqnorm(Er, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo inicial",
cex.main=0.8)
qqline(Er)
qqnorm(Er2, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
cex.main=0.8)
qqline(Er2)
qqnorm(Er3, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
cex.main=0.8)
qqline(Er3)
```

```
# 7.a.2- Ajustados vs. residuales
```

```
layout(matrix(1:3, 1,3))
plot(x=Fit, y=Er, ylim=c(-2.5, 2.5), main="Modelo inicial", cex.main=0.8,
xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

```
plot(x=Fit2, y=Er2, ylim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
cex.main=0.8, xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

```
Fit3<-fitted(mod3)
```

```
plot(x=Fit3, y=Er3, ylim=c(-2.5, 2.5), main="Modelo VarIdent(N*S)",
cex.main=0.8, xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

```
# En este caso, observamos un patrón de distribución similar al modelo inicial.
```

```
# 7.a.3- Ajuste y comparación de los modelos
```

```
library("bbmle")
```

```

AICtab(mod, mod2, mod3, weights = T, delta = TRUE, base=T, sort = TRUE)
  AIC    dAIC  df weight
mod2 810.5    0.0 13 0.870
mod   814.5    4.0 10 0.116
mod3  818.8    8.3 13 0.014

# Comparamos con el modelo inicial
anova(mod, mod3)
  Model df      AIC      BIC    logLik  Test  L.Ratio p-value
mod     1 10 814.4985 831.3873 -397.2493
mod3    2 13 818.8040 840.7595 -396.4020 1 vs 2 1.694499 0.6382

```

# NO mejoró el ajuste respecto del modelo inicial.

### # 7.b- Modelamos las varianzas a nivel de PMS\_gr

```

mod4<-lme(Hord_B~N*S*PMS_gr, random=~1|Localidad,
          weights = varExp(form=~PMS_gr), data=horde)

```

# **TAREA:** escribir al menos dos modelos alternativos posibles utilizando distintas funciones de la varianza.

#### # 7.b.1- Verificación de los modelos

# - Varianzas

# Residuales vs. predictores

# Localidad

```

layout(matrix(1:4, 2,2))

```

```

boxplot(Er~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo inicial",
        cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")

```

```

boxplot(Er2~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
        cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")

```

```

boxplot(Er3~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo VarIdent(N*S)",
        cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")

```

```

Er4<-resid(mod4, type="normalized")

```

```

boxplot(Er4~horde$Localidad, ylim=c(-2.5, 2.5), main="Modelo VarExp(PMS)",
        cex.main=0.8, xlab="Localidad", ylab="Residuales estandarizados")

```



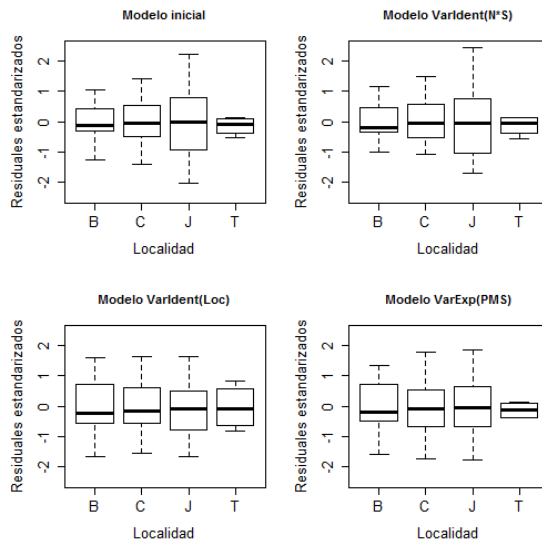


Figura 6. 3: residuales normalizados en relación a la Localidad para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad (VarIdent(Localidad), c) con una función que corrige para la interacción N\*S (VarIdent(N\*S) y, d) con una función exponencial para el peso de mil semillas (VarExp(PMS)).

```
tapply(Er,horde$Localidad,sd)
tapply(Er2,horde$Localidad,sd)
tapply(Er3,horde$Localidad,sd)
tapply(Er4,horde$Localidad,sd)
```

```
# Nitrogeno: Azufre
layout(matrix(1:4, 2,2))
boxplot(Er~horde$N:horde$$, ylim=c(-2.5, 2.5), main="Modelo inicial",
cex.main=0.8, xlab="N*S", ylab="Residuales estandarizados")
boxplot(Er2~horde$N:horde$$, ylim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
cex.main=0.8, xlab="N*S", ylab="Residuales estandarizados")
boxplot(Er3~horde$N:horde$$, ylim=c(-2.5, 2.5), main="Modelo VarIdent(N*S)",
cex.main=0.8, xlab="N*S", ylab="Residuales estandarizados")
boxplot(Er4~horde$N:horde$$, ylim=c(-2.5, 2.5), main="Modelo VarExp(PMS)",
cex.main=0.8, xlab="N*S", ylab="Residuales estandarizados")
```



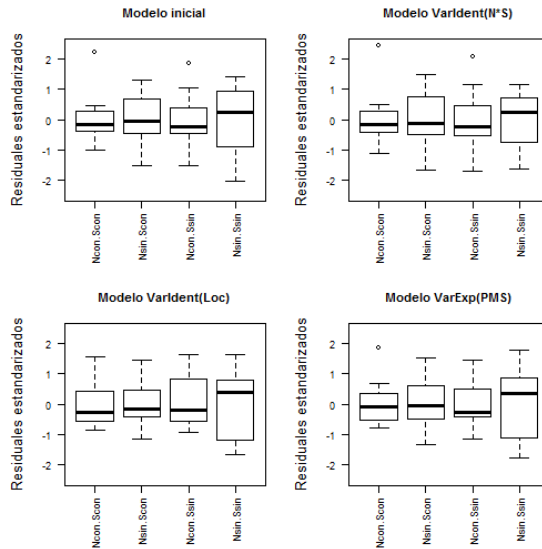


Figura 6. 4: residuales normalizados en relación a la interacción N\*S para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad (VarIdent(Localidad), c) con una función que corrige para la interacción N\*S (VarIdent(N\*S) y, d) con una función exponencial para el peso de mil semillas (VarExp(PMS)).

```
tapply(Er,horde$N:horde$$,sd)
tapply(Er2,horde$N:horde$$,sd)
tapply(Er3,horde$N:horde$$,sd)
tapply(Er4,horde$N:horde$$,sd)
```

```
# Peso de mil semillas (PMS_gr)
layout(matrix(1:4, 2,2))
plot(Er~horde$PMS_gr, ylim=c(-2.5, 2.5), main="Modelo inicial", cex.main=0.8,
xlab="Peso mil semillas", ylab="Residuales estandarizados")
abline(a=0,b=0, col="violet", lw=3)
plot(Er2~horde$PMS_gr, ylim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
cex.main=0.8, xlab="Peso mil semillas", ylab="Residuales estandarizados")
abline(a=0,b=0, col="violet", lw=3)
plot(Er3~horde$PMS_gr, ylim=c(-2.5, 2.5), main="Modelo VarIdent(N*S)",
cex.main=0.8, xlab="Peso mil semillas", ylab="Residuales estandarizados")
abline(a=0,b=0, col="violet", lw=3)
plot(Er4~horde$PMS_gr, ylim=c(-2.5, 2.5), main="Modelo VarExp(PMS)",
cex.main=0.8, xlab="Peso mil semillas", ylab="Residuales estandarizados")
abline(a=0,b=0, col="violet", lw=3)
```

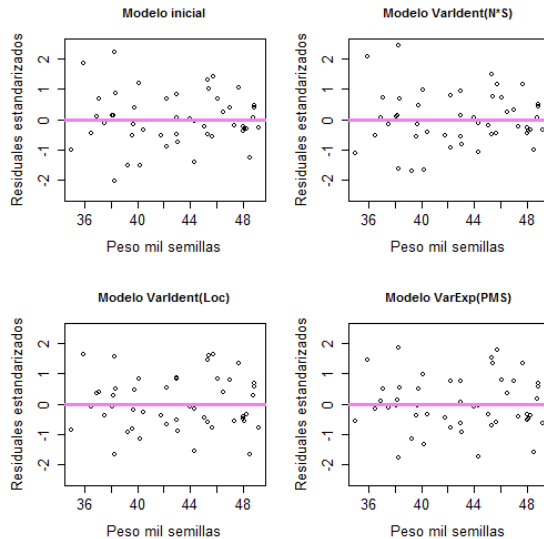


Figura 6. 5: residuales normalizados en relación a la interacción N\*S para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad (VarIdent(Localidad), c) con una función que corrige para la interacción N\*S (VarIdent(N\*S) y, d) con una función exponencial para el peso de mil semillas (VarExp(PMS)).

# - Normalidad

```
layout(matrix(1:4,2,2))
```

```
qqnorm(Er, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo inicial",  
cex.main=0.8)
```

```
qqline(Er)
```

```
qqnorm(Er2, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",  
cex.main=0.8)
```

```
qqline(Er2)
```

```
qqnorm(Er3, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo VarIdent(N*S)",  
cex.main=0.8)
```

```
qqline(Er3)
```

```
qqnorm(Er4, ylim=c(-2.5, 2.5), xlim=c(-2.5, 2.5), main="Modelo VarExp(PMS)",  
cex.main=0.8)
```

```
qqline(Er4)
```

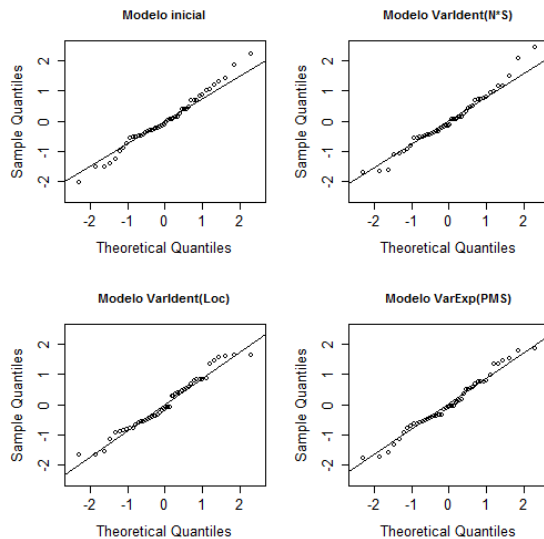


Figura 6. 6: gráficos de normalidad para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad (VarIdent(Localidad), c) con una función que corrige para la interacción N\*S (VarIdent(N\*S) y, d) con una función exponencial para el peso de mil semillas (VarExp(PMS)).

#### # 7.b.2- Ajustados vs. residuales

```
layout(matrix(1:4, 2,2))
```

```
plot(x=Fit, y=Er, ylim=c(-2.5, 2.5), main="Modelo inicial", cex.main=0.8,
xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

```
plot(x=Fit2, y=Er2, ylim=c(-2.5, 2.5), main="Modelo VarIdent(Loc)",
cex.main=0.8, xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

```
plot(x=Fit3, y=Er3, ylim=c(-2.5, 2.5), main="Modelo VarIdent(N*S)",
cex.main=0.8, xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

```
Fit4<-fitted(mod4)
```

```
plot(x=Fit4, y=Er3, ylim=c(-2.5, 2.5), main="Modelo VarExp(PMS)",
cex.main=0.8, xlab="Ajustados", ylab="Residuales normalizados")
abline(a=0,b=0, col="violet", lw=3)
```

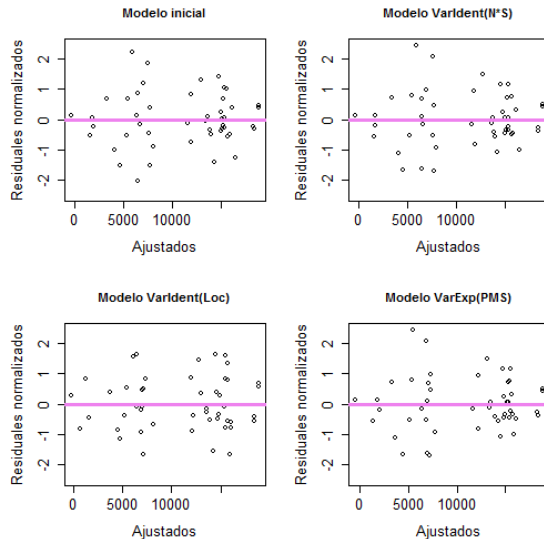


Figura 6. 7: residuales normalizados en relación a los valores ajustados para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad (VarIdent(Localidad), c) con una función que corrige para la interacción N\*S (VarIdent(N\*S) y, d) con una función exponencial para el peso de mil semillas (VarExp(PMS)).

# En este caso, observamos un patrón de distribución similar al modelo inicial.

### # 7.b.3- Ajuste y comparación de los modelos

```
library("bbmle")
```

```
AICTab(mod, mod2, mod3, mod4, weights = T, delta = TRUE, base=T, sort = TRUE)
```

```

AIC    dAIC   df weight
mod2  810.5    0.0  13 0.6097
mod4  811.9    1.4  11 0.2994
mod   814.5    4.0  10 0.0815
mod3  818.8    8.3  13 0.0095

```

```
# Comparamos con el modelo inicial
```

```
anova(mod, mod4)
```

```

Model df      AIC      BIC    logLik  Test  L.Ratio p-value
mod     1  10  814.4985  831.3873 -397.2493
mod4    2  11  811.8951  830.4728 -394.9476 1 vs 2  4.603403  0.0319

```

```
# Mejoró el ajuste respecto del modelo inicial.
```

```
anova(mod2, mod4)
```

```

Model df      AIC      BIC    logLik  Test  L.Ratio p-value
mod2    1  13  810.4730  832.4285 -392.2365
mod4    2  11  811.8951  830.4728 -394.9476 1 vs 2  5.422079  0.0665

```

```
# Pero no, respecto del modelo "mod2"
```

### # 8- Ajustar el componente fijo

```
# Recordar: para este punto existen distintos marcos conceptuales y
```

```
# procedimientos (p.ej.: tabla de anova del modelo, cocientes de verosimilitud (LRT), Inferencia Multimodelo (MMI), etc.).
```

```
# Cualquiera de estos procedimientos es correcto si se lo utiliza adecuadamente.
```

**# 8.a- ANOVA**

# Dado que se trata de un experimento manipulativo, una de las alternativas es utilizar el marco inferencial de un ANOVA

```
anova(mod2)
      (Intercept)  numDF denDF  F-value  p-value
N                1      37  8.111683  0.0071
S                1      37  4.155251  0.0487
PMS_gr           1      37  0.365544  0.5491
N:S              1      37  5.157408  0.0291
N:PMS_gr         1      37  3.351077  0.0752
S:PMS_gr         1      37  0.154651  0.6964
N:S:PMS_gr       1      37  3.629162  0.0646
N:S:PMS_gr       1      37  0.113733  0.7378
```

# La tabla de anova muestra los factores, grados de libertad de numerador y denominador, el estadístico (F-value) y el valor de probabilidad asociado (p-value).

# En este caso, los valores de probabilidad indican que las interacciones no son importantes. Mientras tanto habría un efecto dado por el peso (PMS\_gr) y los tratamientos de fertilización nitrogenada (N).

**# 8.b- Test de cociente de verosimilitud (LRT)**

# (Recordar CAPÍTULO 4)

# Estimar el modelo "mod2" que tuvo el mejor ajuste con el método de máxima verosimilitud (ML)

# Para esto utilizamos la función "update"

```
mod2ML<-update(mod2, method="ML")
```

# El nuevo modelo

```
mod2ML$call
```

```
lme.formula(fixed = Hord_B ~ N * S * PMS_gr, data = horde, random =
~1 | Localidad, weights = varIdent(form = ~1 | Localidad), method =
"ML")
```

**# 8.b.1- Revisar los efectos fijos**

```
summary(mod2ML)$tTable
```

	Value	Std.Error	DF	t-value	p-value
(Intercept)	-10446.067	6702.7333	37	-1.558	0.1276
NNsin	4047.325	13366.4683	37	0.302	0.7637
SSsin	11944.197	6846.6319	37	1.744	0.0893
PMS_gr	485.786	142.6365	37	3.405	0.0016
NNsin:SSsin	-755.015	18135.3948	37	-0.041	0.9670
NNsin:PMS_gr	-162.246	293.7603	37	-0.552	0.5840
SSsin:PMS_gr	-301.875	162.3297	37	-1.859	0.0709
NNsin:SSsin:PMS_gr	91.622	402.5410	37	0.227	0.8212

**# 8.b.2- Comenzar sacando la interacción más compleja: triple**

# Para esto utilizamos la función "update"

```
mod2ML2<-update(mod2ML, ~.-N:S:PMS_gr)
```

**# 8.b.3- Comparar los dos modelos**

```
anova(mod2ML, mod2ML2)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
mod2ML	1	13	923.26	947.59	-448.63			
mod2ML2	2	12	921.32	943.77	-448.66	1 vs 2	0.0588	0.8083

# El LRT indica que no hay diferencias en el ajuste del modelo con o sin la interacción.

# Por parsimonia, excluirémos la interacción del modelo.

# Dado que no hay otra interacción triple,

# la sacamos del modelo y continuamos sacando las interacciones dobles de a una.

#### *# 8.b.4- Sacamos de a una las interacciones dobles y comparamos los modelos*

```
summary(mod2ML2)$tTable
```

```
mod2ML3<-update(mod2ML2, ~.-S:PMS_gr)
```

```
anova(mod2ML2, mod2ML3)
```

# El test indica que el modelo son la interacción ajusta significativamente peor que el modelo con la interacción.

```
mod2ML4<-update(mod2ML2, ~.-N:PMS_gr)
```

```
anova(mod2ML2, mod2ML4)
```

```
mod2ML5<-update(mod2ML2, ~.-N:S)
```

```
anova(mod2ML2, mod2ML5)
```

```
# Idem
```

# Dado que dos de las interacciones son significativas, en este caso, el nuevo modelo sería "mod2ML4"

```
mod2ML4$call
```

# Dado que las dos interacciones que persisten en el modelo incluyen a los tres factores simples, no podemos sacarlos.

#### *# 8.b.5- Tabla resumen del LRT*

```
Termino<-c("N:S:PMS_gr",
           "",
           "S:PMS_gr",
           "N:PMS_gr",
           "N:S")
```

```
LRT<-c(round(anova(mod2ML, mod2ML2)[2,"L.Ratio"],4),
        "",
        round(anova(mod2ML2, mod2ML3)[2,"L.Ratio"],4),
        round(anova(mod2ML2, mod2ML4)[2,"L.Ratio"],4),
        round(anova(mod2ML2, mod2ML5)[2,"L.Ratio"],4))
```

```
gl<-c(summary(mod2ML)$tTable["NNsin:SSsin:PMS_gr", "DF"],
        "",
        summary(mod2ML)$tTable["SSsin:PMS_gr", "DF"],
        summary(mod2ML)$tTable["SSsin:PMS_gr", "DF"],
        summary(mod2ML)$tTable["NNsin:SSsin", "DF"])
```

```
p_value<-c(round(anova(mod2ML, mod2ML2)[2,"p-value"],3),
            "",
            round(anova(mod2ML2, mod2ML3)[2,"p-value"],3),
```

```
round(anova(mod2ML2, mod2ML4)[2,"p-value"],3),
round(anova(mod2ML2, mod2ML5)[2,"p-value"],3))
```

```
Table<-data.frame(Termino,LRT,gl,p_value)
```

```
Table
  Termino  LRT gl p_value
1 N:S:PMS_gr 0.0589 37 0.808
2
3 S:PMS_gr 4.1721 37 0.041
4 N:PMS_gr 0.3134 37 0.576
5 N:S 6.2347 37 0.013
```

### # 8.c - Función Anova

```
# Comparamos los resultados del LRT anterior con la función Anova (car)
```

```
# library(car)
```

```
# Anova(mod2ML)
```

```
Analysis of Deviance Table (Type II tests)
```

```
Response: Hord_B
```

```
      Chisq Df Pr(>Chisq)
N      5.6410  1  0.017545 *
S      0.0074  1  0.931412
PMS_gr  9.6497  1  0.001894 **
N:S     7.0218  1  0.008052 **
N:PMS_gr 0.3326  1  0.564124
S:PMS_gr  4.4887  1  0.034119 *
N:S:PMS_gr 0.0622  1  0.803103
```

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

```
# Al comparar el LRT y el ANOVA:
```

```
# - La tabla del ANOVA del modelo "mod2" tampoco encontró un valor
significativo
```

```
# para las interacciones: "N:S:PMS_gr", "N:PMS_gr".
```

```
# - De manera similar al LRT, hubo una probabilidad marginal para las
interacciones "S:PMS_gr" y "N:S".
```

### # 9- Modelo final

#### # 9.a- Presentación del modelo

```
# El modelo final sería el modelo 14 ajustado por el método de REML
```

```
modFinal<-update(mod2ML4, method="REML")
```

```
modFinal$call
```

```
lme.formula(fixed = Hord_B ~ N + S + PMS_gr + N:S + S:PMS_gr,
  data = horde, random = ~1 | Localidad, weights = varIdent(form =
  ~1 |
  Localidad), method = "REML")
```

```
# El modelo se puede representar de manera matemática como:
```

$$Y_i \sim N(\mu_i, \sigma_{\epsilon, localidad}^2)$$

$$\mu_i = \beta_{0j[i]} + \beta_1 * N_i + \beta_2 * S_i + \beta_3 * Peso\_mil_i + \beta_4 * S_i * N_i + \beta_5 * S_i * Peso\_mil_i$$

$$\beta_{0j} \sim N(\mu_{\beta_0}, \sigma_{\beta_0}^2)$$

$$\mu_{\beta_0} = \alpha_{0j}$$

$$i = 1, 2, 3, \dots, 48 \text{ parcelas}$$

$j = 1,2,3,4$  localidades

**Donde,**

$Y_i =$  cantidad de hordeína B en grano de cebada

Coefficientes:

$\beta_{0j} =$  intercepción

$\beta_1 =$  fertilización con nitrógeno

$\beta_2 =$  fertilización con azufre

$\beta_3 =$  peso de mil semillas

$\beta_4 =$  nitrógeno x azufre

$\beta_5 =$  azufre x peso de mil semillas

$\alpha_{0j} =$  intercepción

$\sigma_{\beta_0}^2 =$  varianza entre localidades

$\sigma_{\epsilon}^2 =$  varianza residual

### # 9.b- Gráficos de ajuste del modelo

#### # Errores y ajustados

```
ErFin<-resid(modFinal, type="normalized")
```

```
FitFin<-fitted(modFinal)
```

```
layout(matrix(1:4, 2,2))
```

```
# la función "layout" permite dimensionar la cantidad de gráficos
```

```
# y distribución en la ventana de gráficos
```

```
plot(x=FitFin, y=ErFin, xlab="Ajustados", ylab="Residuales normalizados")
```

```
abline(0,0, col="red", lwd= 2)
```

```
# ¡IMPORTANTE! Siempre deseamos un gráfico que no muestre un patrón o una tendencia.
```

```
boxplot(ErFin~horde$N, main="Fertilización nitrogenada")
```

```
boxplot(ErFin~horde$S, main="Fertilización azufrada")
```

```
plot(ErFin~horde$PMS_gr, main="Peso de mil semillas (g)")
```

```
abline(0,0, col="red", lwd= 2)
```



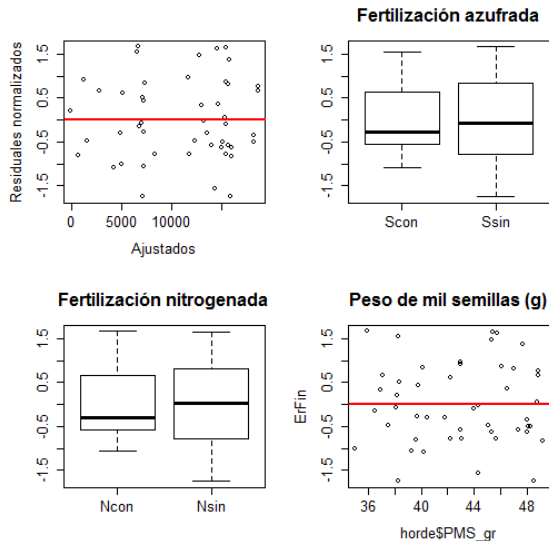


Figura 6. 8: residuales normalizados del modelo "modFinal" en relación a los valores ajustados por el modelo, cada uno de los predictores (fertilización con azufre, fertilización con nitrógeno y peso de mil semillas).

#### # Ajustados vs. observados

```
layout(matrix(1:1, 1,1))
plot(FitFin,horde$Hord_B, xlab="Observados", ylab="Ajustados")
# agregamos una línea con ordenada al origen = 0 y pendiente = 1
abline(0,1,col="red", lwd= 2)
```

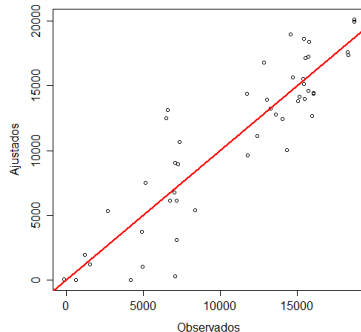


Figura 6. 9: valores ajustados por el modelo "modFinal" en relación a los valores de horteínas B observados.

#### # 10- Graficar el modelo promedio general

```
cl<-with(horde, length(levels(interaction(N,S))))
horde$trat<-interaction(horde$N,horde$S)
```

```
layout(matrix(1:1,1,1))
plot(x=horde$PMS_gr, y = horde$Hord_B,
     type = "n",lwd=2 ,col="gray",
     ylab="Cantidad de horteínas B", xlab="Peso de mil semillas (g)")
```

# type="n" indica que el gráfico esta vacío.

# A continuación agregamos los puntos para cada tratamiento de fertiización con distintos colores

```
# Componentes fijos
```

```
fixef(modFinal)
```

```
# Ncon-Scon
```

```
points(horde$PMS_gr[horde$trat == "Ncon.Scon"],horde$Hord_B[horde$trat ==
"Ncon.Scon"],
```

```
col = "black", pch = 16)
```

```
curve(fixef(modFinal)[1]+
```

```
fixef(modFinal)[4]*(x),add=T, col="black", lw=2 )
```

```
# Nsin-Scon
```

```
points(horde$PMS_gr[horde$trat == "Nsin.Scon"],horde$Hord_B[horde$trat ==
"Nsin.Scon"],
```

```
col = "red", pch = 16)
```

```
curve(fixef(modFinal)[1]+
```

```
fixef(modFinal)[2]+
```

```
fixef(modFinal)[4]*(x),add=T, col="red", lw=2 )
```

```
# Ncon-Ssin
```

```
points(horde$PMS_gr[horde$trat == "Ncon.Ssin"],horde$Hord_B[horde$trat ==
"Ncon.Ssin"],
```

```
col = "green", pch = 16)
```

```
curve(fixef(modFinal)[1]+
```

```
fixef(modFinal)[3]+
```

```
fixef(modFinal)[4]*(x)+
```

```
fixef(modFinal)[6]*(x),add=T, col="green", lw=2 )
```

```
# Nsin-Ssin
```

```
points(horde$PMS_gr[horde$trat == "Nsin.Ssin"],horde$Hord_B[horde$trat ==
"Nsin.Ssin"],
```

```
col = "blue", pch = 16)
```

```
curve(fixef(modFinal)[1]+
```

```
fixef(modFinal)[2]+
```

```
fixef(modFinal)[3]+
```

```
fixef(modFinal)[5]+
```

```
fixef(modFinal)[4]*(x)+
```

```
fixef(modFinal)[6]*(x),add=T, col="blue", lw=2 )
```

```
legend("topleft",levels(interaction(horde$N,horde$S)), col=1:cl, pch=16,
```

```
bty="n")
```

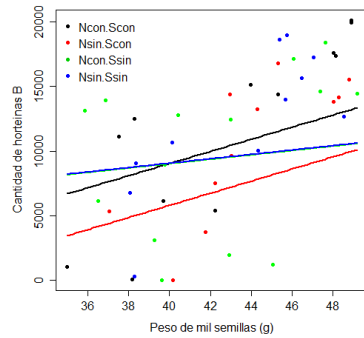


Figura 6. 10: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g) y ajuste general del modelo (líneas llenas). Los colores indican los tratamientos NconScon (negro), NsinScon (rojo), NconSsin (verde), NsinSsin (azul).

### # Observar:

# - las líneas de ajuste muestran la interacción S:PMS\_gr: Las dos líneas con azufre (negro y rojo) tienen pendiente distinta de las líneas sin azufre (verde y azul)

# - El efecto de la fertilización con nitrógeno, se pone en evidencia al fertilizar con azufre. Esto es, la interacción N:S

### # 11- Graficar los efectos aleatorios

# Graficamos los componentes fijos del modelo general (líneas punteadas) y las intercepciones aleatorias para cada localidad (líneas llenas)

```
cl<-with(horde, length(levels(interaction(N,S))))
horde$trat<-interaction(horde$N,horde$S)
```

```
layout(matrix(1:1,1,1))
plot(x=horde$PMS_gr, y = horde$Hord_B,
     type = "n",lwd=2 ,col="gray",
     ylab="Cantidad de horteinas B", xlab="Peso de mil semillas (g)",
     main="Baigorrita")
```

# A continuación agregamos los puntos para cada tratamiento de fertilización con distintos colores

### # Componentes fijos

```
fixef(modFinal)
```

# Componentes aleatorios

```
ranef(modFinal)
```

### # Ncon-Scon

```
# points(horde$PMS_gr[horde$trat == "Ncon.Scon"],horde$Hord_B[horde$trat == "Ncon.Scon"],
```

```
#   col = "black", pch = 16)
```

```
curve(fixef(modFinal)[1]+
```

```
      fixef(modFinal)[4]*(x),add=T, col="black", lw=1, lty="dashed")
```

### # Nsin-Scon

```

# points(horde$PMS_gr[horde$trat == "Nsin.Scon"],horde$Hord_B[horde$trat
== "Nsin.Scon"],
#   col = "red", pch = 16)
curve(fixef(modFinal)[1]+
      fixef(modFinal)[2]+
      fixef(modFinal)[4]*(x),add=T, col="red", lw=1, lty="dashed")
# Ncon-Ssin
# points(horde$PMS_gr[horde$trat == "Ncon.Ssin"],horde$Hord_B[horde$trat
== "Ncon.Ssin"],
#   col = "green", pch = 16)
curve(fixef(modFinal)[1]+
      fixef(modFinal)[3]+
      fixef(modFinal)[4]*(x)+
      fixef(modFinal)[6]*(x),add=T, col="green", lw=1, lty="dashed")
# Nsin-Ssin
# points(horde$PMS_gr[horde$trat == "Nsin.Ssin"],horde$Hord_B[horde$trat ==
"Nsin.Ssin"],
#   col = "blue", pch = 16)
curve(fixef(modFinal)[1]+
      fixef(modFinal)[2]+
      fixef(modFinal)[3]+
      fixef(modFinal)[5]+
      fixef(modFinal)[4]*(x)+
      fixef(modFinal)[6]*(x),add=T, col="blue", lw=1,lty="dashed")

# Baigorrita
ranef(modFinal)
ranef(modFinal)[1,1]
# Ncon-Scon
points(horde$PMS_gr[horde$trat ==
"Ncon.Scon"&horde$Localidad=="B"],horde$Hord_B[horde$trat ==
"Ncon.Scon"&horde$Localidad=="B"],
      col = "black", pch = 16)
curve(ranef(modFinal)[1,1]+fixef(modFinal)[1]+
      fixef(modFinal)[4]*(x),add=T, col="black", lw=2 )

# Nsin-Scon
points(horde$PMS_gr[horde$trat ==
"Nsin.Scon"&horde$Localidad=="B"],horde$Hord_B[horde$trat ==
"Nsin.Scon"&horde$Localidad=="B"],
      col = "red", pch = 16)
curve(ranef(modFinal)[1,1]+fixef(modFinal)[1]+
      fixef(modFinal)[2]+
      fixef(modFinal)[4]*(x),add=T, col="red", lw=2 )

# Ncon-Ssin
points(horde$PMS_gr[horde$trat ==
"Ncon.Ssin"&horde$Localidad=="B"],horde$Hord_B[horde$trat ==
"Ncon.Ssin"&horde$Localidad=="B"],

```

```
col = "green", pch = 16)
curve(ranef(modFinal)[1,1]+fixef(modFinal)[1]+
      fixef(modFinal)[3]+
      fixef(modFinal)[4]*(x)+
      fixef(modFinal)[6]*(x),add=T, col="green", lw=2 )
```

**# Nsin-Ssin**

```
points(horde$PMS_gr[horde$trat ==
"Nsin.Ssin"&horde$Localidad=="B"],horde$Hord_B[horde$trat ==
"Nsin.Ssin"&horde$Localidad=="B"],
      col = "blue", pch = 16)
curve(ranef(modFinal)[1,1]+ fixef(modFinal)[1]+
      fixef(modFinal)[2]+
      fixef(modFinal)[3]+
      fixef(modFinal)[5]+
      fixef(modFinal)[4]*(x)+
      fixef(modFinal)[6]*(x),add=T, col="blue", lw=2 )
```

```
legend("bottomright",levels(interaction(horde$N,horde$S)), col=1:cl, pch=16,
      bty="n")
```

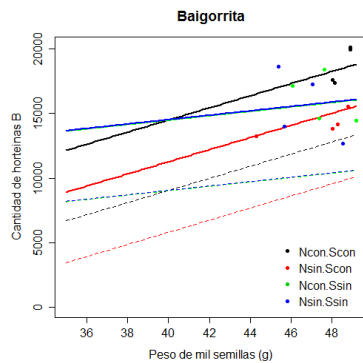


Figura 6. 11: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g), ajuste general del modelo (líneas punteadas) y, ajuste considerando la interceptión aleatoria para la localidad de Baigorrita (líneas llenas). Los colores indican los tratamientos NconScon (negro), NsinScon (rojo), NconSsin (verde), NsinSsin (azul).

```
#=====
plot(x=horde$PMS_gr, y = horde$Hord_B,
      type = "n",lwd=2 ,col="gray",
      ylab="Cantidad de horteinas B", xlab="Peso de mil semillas (g)",
      main="Campito")
# A continuación agregamos los puntos para cada tratamiento de fertiización con
distintos colores
```

**# Ncon-Scon**

```
# points(horde$PMS_gr[horde$trat == "Ncon.Scon"],horde$Hord_B[horde$trat
== "Ncon.Scon"],
#   col = "black", pch = 16)
curve(fixef(modFinal)[1]+
      fixef(modFinal)[4]*(x),add=T, col="black", lw=1, lty="dashed")
```

**# Nsin-Scon**

```
curve(fixef(modFinal)[1]+
      fixef(modFinal)[2]+
      fixef(modFinal)[4]*(x),add=T, col="red", lw=1, lty="dashed")
```

**# Ncon-Ssin**

```
curve(fixef(modFinal)[1]+
      fixef(modFinal)[3]+
      fixef(modFinal)[4]*(x)+
      fixef(modFinal)[6]*(x),add=T, col="green", lw=1, lty="dashed")
```

**# Nsin-Ssin**

```
curve(fixef(modFinal)[1]+
      fixef(modFinal)[2]+
      fixef(modFinal)[3]+
      fixef(modFinal)[5]+
      fixef(modFinal)[4]*(x)+
      fixef(modFinal)[6]*(x),add=T, col="blue", lw=1,lty="dashed")
```

**# Campito**

```
ranef(modFinal)
ranef(modFinal)[2,1]
```

**# Ncon-Scon**

```
points(horde$PMS_gr[horde$trat ==
"Ncon.Scon"&horde$Localidad=="C"],horde$Hord_B[horde$trat ==
"Ncon.Scon"&horde$Localidad=="C"],
      col = "black", pch = 16)
curve(ranef(modFinal)[2,1]+fixef(modFinal)[1]+
      fixef(modFinal)[4]*(x),add=T, col="black", lw=2 )
```

**# Nsin-Scon**

```
points(horde$PMS_gr[horde$trat ==
"Nsin.Scon"&horde$Localidad=="C"],horde$Hord_B[horde$trat ==
"Nsin.Scon"&horde$Localidad=="C"],
      col = "red", pch = 16)
curve(ranef(modFinal)[2,1]+fixef(modFinal)[1]+
      fixef(modFinal)[2]+
      fixef(modFinal)[4]*(x),add=T, col="red", lw=2 )
```

**# Ncon-Ssin**

```
points(horde$PMS_gr[horde$trat ==
"Ncon.Ssin"&horde$Localidad=="C"],horde$Hord_B[horde$trat ==
"Ncon.Ssin"&horde$Localidad=="C"],
      col = "green", pch = 16)
curve(ranef(modFinal)[2,1]+fixef(modFinal)[1]+
      fixef(modFinal)[3]+
      fixef(modFinal)[4]*(x)+
      fixef(modFinal)[6]*(x),add=T, col="green", lw=2 )
```

**# Nsin-Ssin**

```

points(horde$PMS_gr[horde$trat ==
"Nsin.Ssin"&horde$Localidad=="C"],horde$Hord_B[horde$trat ==
"Nsin.Ssin"&horde$Localidad=="C"],
  col = "blue", pch = 16)
curve(ranef(modFinal)[2,1]+ fixef(modFinal)[1]+
  fixef(modFinal)[2]+
  fixef(modFinal)[3]+
  fixef(modFinal)[5]+
  fixef(modFinal)[4]*(x)+
  fixef(modFinal)[6]*(x),add=T, col="blue", lw=2 )

```

```

legend("bottomright",levels(interaction(horde$N,horde$S)), col=1:4, pch=16,
bty="n")

```

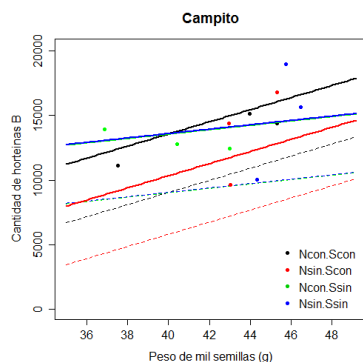


Figura 6. 12: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g), ajuste general del modelo (líneas punteadas) y, ajuste considerando la interceptación aleatoria para la localidad de Campito (líneas llenas). Los colores indican los tratamientos NconScon (negro), NsinScon (rojo), NconSsin (verde), NsinSsin (azul).

```

#=====
plot(x=horde$PMS_gr, y = horde$Hord_B,
  type = "n",lwd=2 ,col="gray",
  ylab="Cantidad de horteinas B", xlab="Peso de mil semillas (g)",
  main="Junin")
# A continuación agregamos los puntos para cada tratamiento de fertilización
con distintos colores

# Ncon-Scon
# points(horde$PMS_gr[horde$trat == "Ncon.Scon"],horde$Hord_B[horde$trat
== "Ncon.Scon"],
#   col = "black", pch = 16)
curve(fixef(modFinal)[1]+
  fixef(modFinal)[4]*(x),add=T, col="black", lw=1, lty="dashed")
# Nsin-Scon
curve(fixef(modFinal)[1]+
  fixef(modFinal)[2]+
  fixef(modFinal)[4]*(x),add=T, col="red", lw=1, lty="dashed")
# Ncon-Ssin
curve(fixef(modFinal)[1]+

```

```

fixef(modFinal)[3]+
fixef(modFinal)[4]*(x)+
fixef(modFinal)[6]*(x),add=T, col="green", lw=1, lty="dashed")

```

#### # Nsin-Ssin

```

curve(fixef(modFinal)[1]+
fixef(modFinal)[2]+
fixef(modFinal)[3]+
fixef(modFinal)[5]+
fixef(modFinal)[4]*(x)+
fixef(modFinal)[6]*(x),add=T, col="blue", lw=1,lty="dashed")

```

#### # Junin

```

ranef(modFinal)
ranef(modFinal)[3,1]

```

#### # Ncon-Scon

```

points(horde$PMS_gr[horde$trat ==
"Ncon.Scon"&horde$Localidad=="J"],horde$Hord_B[horde$trat ==
"Ncon.Scon"&horde$Localidad=="J"],
col = "black", pch = 16)
curve(ranef(modFinal)[3,1]+fixef(modFinal)[1]+
fixef(modFinal)[4]*(x),add=T, col="black", lw=2 )

```

#### # Nsin-Scon

```

points(horde$PMS_gr[horde$trat ==
"Nsin.Scon"&horde$Localidad=="J"],horde$Hord_B[horde$trat ==
"Nsin.Scon"&horde$Localidad=="J"],
col = "red", pch = 16)
curve(ranef(modFinal)[3,1]+fixef(modFinal)[1]+
fixef(modFinal)[2]+
fixef(modFinal)[4]*(x),add=T, col="red", lw=2 )

```

#### # Ncon-Ssin

```

points(horde$PMS_gr[horde$trat ==
"Ncon.Ssin"&horde$Localidad=="J"],horde$Hord_B[horde$trat ==
"Ncon.Ssin"&horde$Localidad=="J"],
col = "green", pch = 16)
curve(ranef(modFinal)[3,1]+fixef(modFinal)[1]+
fixef(modFinal)[3]+
fixef(modFinal)[4]*(x)+
fixef(modFinal)[6]*(x),add=T, col="green", lw=2 )

```

#### # Nsin-Ssin

```

points(horde$PMS_gr[horde$trat ==
"Nsin.Ssin"&horde$Localidad=="J"],horde$Hord_B[horde$trat ==
"Nsin.Ssin"&horde$Localidad=="J"],
col = "blue", pch = 16)
curve(ranef(modFinal)[3,1]+ fixef(modFinal)[1]+

```



```
fixef(modFinal)[2]+
fixef(modFinal)[3]+
fixef(modFinal)[5]+
fixef(modFinal)[4]*(x)+
fixef(modFinal)[6]*(x),add=T, col="blue", lw=2 )
```

```
legend("bottomright",levels(interaction(horde$N,horde$S)), col=1:cl, pch=16,
bty="n")
```

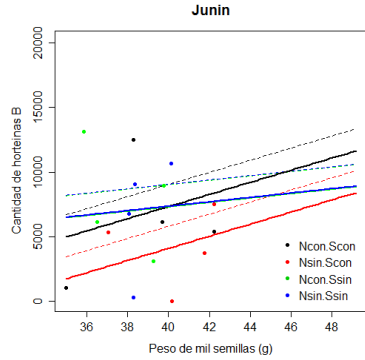


Figura 6. 13: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g), ajuste general del modelo (líneas punteadas) y, ajuste considerando la interceptión aleatoria para la localidad de Junín (líneas llenas). Los colores indican los tratamientos NconScon (negro), NsinScon (rojo), NconSsin (verde), NsinSsin (azul).

```
#=====
plot(x=horde$PMS_gr, y = horde$Hord_B,
     type = "n",lwd=2 ,col="gray",
     ylab="Cantidad de horteinas B", xlab="Peso de mil semillas (g)",
     main="Tiburcio")
# A continuación agregamos los puntos para cada tratamiento de fertilización
con distintos colores

# Ncon-Scon
# points(horde$PMS_gr[horde$trat == "Ncon.Scon"],horde$Hord_B[horde$trat
== "Ncon.Scon"],
#   col = "black", pch = 16)
curve(fixef(modFinal)[1]+
      fixef(modFinal)[4]*(x),add=T, col="black", lw=1, lty="dashed")
# Ncon-Ssin
curve(fixef(modFinal)[1]+
      fixef(modFinal)[3]+
      fixef(modFinal)[4]*(x)+
      fixef(modFinal)[6]*(x),add=T, col="green", lw=1, lty="dashed")

# Tiburcio
ranef(modFinal)
ranef(modFinal)[4,1]

# Ncon-Scon
```

```
points(horde$PMS_gr[horde$trat ==
"Ncon.Scon"&horde$Localidad=="T"],horde$Hord_B[horde$trat ==
"Ncon.Scon"&horde$Localidad=="T"],
  col = "black", pch = 16)
curve(ranef(modFinal)[4,1]+fixef(modFinal)[1]+
  fixef(modFinal)[4]*(x),add=T, col="black", lw=2 )
```

### # Ncon-Ssin

```
points(horde$PMS_gr[horde$trat ==
"Ncon.Ssin"&horde$Localidad=="T"],horde$Hord_B[horde$trat ==
"Ncon.Ssin"&horde$Localidad=="T"],
  col = "green", pch = 16)
curve(ranef(modFinal)[4,1]+fixef(modFinal)[1]+
  fixef(modFinal)[3]+
  fixef(modFinal)[4]*(x)+
  fixef(modFinal)[6]*(x),add=T, col="green", lw=2 )
```

```
legend("topright",c("Ncon.Scon","Ncon.Ssin"), col=c("black", "green"), pch=16,
  bty="n")
```

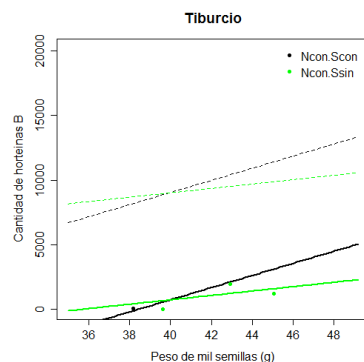


Figura 6. 14: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g), ajuste general del modelo (líneas punteadas) y, ajuste considerando la intercepción aleatoria para la localidad de Tiburcio (líneas llenas). Los colores indican los tratamientos NconScon (negro), NconSsin (verde).

### # ¡ATENCIÓN!

# El conjunto de datos original, tiene solo cuatro datos para la Localidad Tiburcio:

# - tres para el tratamientos Ncon.Scon y uno para el tratamiento Ncon.Ssin.

```
summary(subset(horde, Localidad=="T"))
```

```
which(horde$Localidad=="T")
```

```
horde[horde$Localidad=="T",]
```

	Parcela	N	S	PMS_gr	Hord_B	Hord_C	Hord_D	Hord_Total	trat
2	T2	Ncon	Ssin	39.6	0.0	126.9	0	126.9	Ncon.Ssin
5	T4	Ncon	Scon	38.1	51.2	0.0	0	51.2	Ncon.Scon
24	T18	Ncon	Ssin	42.9	1955.5	0.0	0	1955.5	Ncon.Ssin
26	T20	Ncon	Ssin	45.0	1179.6	0.0	0	1179.6	Ncon.Ssin

### # IMPORTANTE:

# Una de las ventajas de los modelos mixtos ("partial pooling" que discutimos en el capítulo 4) es que a pesar de tener pocos datos, pudimos incluirlos en el

análisis y tener una estimación para esa localidad ponderada por los datos de las otras tres localidades.

### # 12- Respuesta a la pregunta

# ¿Qué influencia tienen el Nitrógeno y el Azufre en la producción de hordeínas en grano de cebada?

# El modelo sugiere que el azufre no tuvo influencia sobre la cantidad de hordeína B

# en cebada, el efecto del nitrógeno y el peso de mil semillas.....

# Para continuar discutiendo.....

### # 13- Agradecimiento

# Datos gentilmente provistos por Andrés Peton y Eduardo Pagano de la Facultad de Agronomía de la Universidad de Buenos Aires. Peton, A. "Patrón de acumulación de hordeínas en granos de cebada cervecera y su relación con la fertilización azufrada y nitrogenada". Tesis de maestría en curso, EPG-FAUBA. Por razones didácticas, los datos presentados son un subconjunto modificado de los datos originales.

# Medidas repetidas con intervalos iguales

## Contenidos

<b># CAPÍTULO 7: Autocorrelación - Medidas repetidas con intervalos iguales....</b>	<b>109</b>
<b># 1- Configuración inicial.....</b>	<b>109</b>
# 1.a- Definir el directorio de trabajo .....	109
# 1.b- Leer el archivo de datos .....	109
<b># 2- El caso: Plasticidad fenotípica de <i>Baccharis medullosa</i> y <i>Eupatorium buniifolium</i> .....</b>	<b>109</b>
# 2.a- Pregunta de interés .....	109
# 2.b- Variable respuesta.....	109
# 2.c- Variable/s predictoras de interés.....	109
# 2.d- Diseño del experimento .....	110
<b># 3- Identificar jerarquías .....</b>	<b>110</b>
<b># 4- Explorar los datos .....</b>	<b>110</b>
# Exploramos la especie de interés: <i>Baccharis medullosa</i> .....	110
<b># 5- Construcción del modelo inicial .....</b>	<b>111</b>
# 5.a- Componente fija .....	111
# 5.b- Componente aleatoria.....	111
<b># 6- Verificación del modelo.....</b>	<b>112</b>
<b># 7- AUTOCORRELACIÓN.....</b>	<b>112</b>
# 7.a- Evaluar la correlación temporal entre los residuos.....	112
# 7.b- Modelos que incluyen funciones de correlación .....	113
<b># 8- Comparamos los modelos.....</b>	<b>114</b>
# Verificamos la validez .....	115
<b># 9- Varianzas heterogéneas .....</b>	<b>115</b>
# 9.a- Modelamos las varianzas .....	115
# 9.b- Comparamos los modelos.....	116
# 9.c- Evaluar la presencia de autocorrelación en este nuevo modelo.....	116
<b># 10- Veamos la información que nos da el modelo.....</b>	<b>117</b>
<b># 11- Inferencia sobre los efectos fijos .....</b>	<b>117</b>
<b># 12- Gráfico del modelo .....</b>	<b>117</b>
<b># 13- Resumen .....</b>	<b>119</b>
<b># 14- Tarea .....</b>	<b>119</b>
<b># 15- Agradecimiento.....</b>	<b>119</b>

## # CAPÍTULO 7: Autocorrelación - Medidas repetidas con intervalos iguales

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo de datos

```
plasti<- read.csv("7_Plasticidad.csv", header = TRUE, na.strings="-9999")
#lee el archivo
summary(plasti)
```

### # 2- El caso: Plasticidad fenotípica de *Baccharis medullosa* y *Eupatorium buniifolium*



Plasticidad en respuesta a diferencias en la textura del suelo y la disponibilidad de agua

#### Especies

*Eupatorium buniifolium*

*Baccharis medullosa*

**Variable respuesta:** Altura

**Sustratos:** Arena y Tierra

**Niveles de riego:** Riego y Sequía

#### Diseño:

2 Sustratos x 2 Niveles de riego  
x 5 fechas x 16 repeticiones  
= 320 datos/sp

Figura 7. 1: esquema del experimento

#### # 2.a- Pregunta de interés

```
# ¿Las condiciones edáficas modifican la altura de B. medullosa?
# (esto fue parte de una pregunta más grande relacionada con plasticidad
# fenotípica y compensación de respuestas a las condiciones edáficas)
# Utilizaremos de ejemplo, una de las especies
```

#### # 2.b- Variable respuesta

```
# Altura de Baccharis medullosa.
```

#### # 2.c- Variable/s predictoras de interés

```
# - Sustratos (Arena [a] y Tierra [t])
# - Niveles de riego (Riego [r] y Sequía [s])
# - Tiempo: 5 fechas a lo largo del experimento
```

### # 2.d- Diseño del experimento

# 2 Sustratos x 2 Niveles de riego x 5 fechas x 16 repeticiones = 320 datos/sp (Fig. 7.1)

### # 3- Identificar jerarquías

# Las escalas y las variables correspondientes a cada una en este caso son en:

# - Macetas [o plantas (una planta por maceta)], donde se aplicaron los tratamientos de Sustrato x Riego

# - Tiempo (medidas repetidas) dentro de cada Maceta: la Altura fue medida a nivel de tiempo

### # 4- Explorar los datos

```
layout(matrix(1:1,1,1))
```

```
cl<-length(levels(plasti$Especie))
```

# Generamos un objeto que contiene la cantidad (length) de niveles (levels) de Especies.

# Este objeto lo usamos en el siguiente gráfico para indicar distintos colores para cada Especie.

```
with(plasti, plot(Fecha,Altura,col=1:cl, pch=16, xlab = "Fecha",ylab = "Altura (cm)"))
```

```
legend("topleft",levels(plasti$Especie), col=1:cl, pch=16, bty="o",bg="light grey" )
```

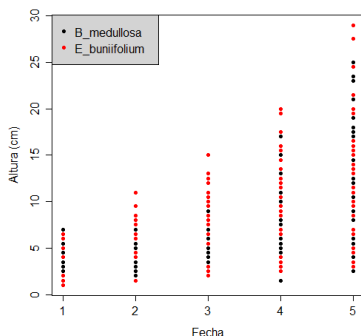


Figura 7. 2: alturas (cm) observadas en el tiempo (Fechas de 1 a 5) de las plantas de *Baccharis medullosa* (puntos negros) y *Eupatorium buniifolium* (puntos rojos).

```
library(lattice)
```

```
with(plasti, xyplot(Altura ~ Fecha|Sustrato*Riego, main="Altura vs Fecha agrupados por riego y sustrato"))
```

### # Exploramos la especie de interés: *Baccharis medullosa*

# Creamos un objeto que solo contiene a la especie que nos interesa:

```
"B_medullosa"
```

```
plasti_Bm<-subset(plasti, Especie=="B_medullosa")
```

```
summary(plasti_Bm)
```

```
names(plasti_Bm)
```

# Consideramos factores a las variables Planta y Fecha

```
plasti_Bm$planta<-as.factor(plasti_Bm$Planta)
```

```
plasti_Bm$fecha<-as.factor(plasti_Bm$Fecha)
```

```
summary(plasti_Bm)
dim(plasti_Bm)
[1] 320 9
```

```
with(plasti_Bm, plot(Fecha, Altura, xlab = "Fecha", ylab = "Altura"))
```

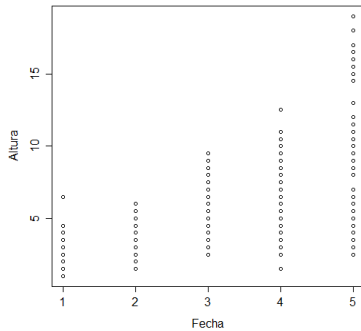


Figura 7. 3: alturas (cm) observadas en el tiempo (Fechas de 1 a 5) de las plantas de *Baccharis medullosa*.

```
with(plasti_Bm, coplot(Altura~Fecha|Sustrato:Riego))
# Este tipo de gráfico también nos ayuda a observar diferencias en los valores y
# en la variabilidad de los datos entre grupos.
```

### # 5- Construcción del modelo inicial

# **ATENCIÓN:** es importante ordenar los datos siguiendo la estructura de jerarquías.  
 # Esto se debe a que la función que estima la autocorrelación entre los datos es empírica.  
 # Es decir, depende del orden en que se encuentran los datos.

```
plasti_Bm<-
plasti_Bm[with(plasti_Bm,order(Especie,Riego,Sustrato,Planta,Fecha)), ]
# Observar los datos para verificar el orden.
# Deberíamos observar las Fechas para una misma planta en filas consecutivas.
View(plasti_Bm)
```

#### # 5.a- Componente fija

# Plantear el modelo con todos los factores de interés y la estructura de jerarquía de los datos.  
 # Incluimos todas las variables e interacciones sin incluir por ahora una función de correlación para las medidas en el tiempo.  
 # A partir de este modelo evaluamos si corresponde o no corregir la autocorrelación temporal.

```
library(nlme)
# Escribimos una función que incluye a todos los factores fijos del modelo
func<-formula(Altura ~ Riego*Sustrato*Fecha)
```

#### # 5.b- Componente aleatoria

# **IMPORTANTE:** la componente aleatoria consiste en el término aleatorio  
 # (que modela la estructura jerárquica del diseño o anidamiento)

# y, términos que permiten modelar heterogeneidad de varianzas (funciones de la varianza), correlación temporal, o correlación espacial.

# Los modelos no necesariamente incluyen todos los componentes aleatorios.

### # Término aleatorio

```
Bm_mod <- lme(func, data = plasti_Bm, random = ~1|fplanta)
```

### # 6- Verificación del modelo

```
layout(matrix(1:4,2,2))
```

```
Fit<- fitted(Bm_mod)
```

```
Er<- residuals(Bm_mod, type="n")
```

```
plot(x=Fit, y=Er, xlab="Ajustados", ylab="Residuales estandarizados")
```

```
abline(a=0, b=0, col="red", lw=2)
```

# Observar el patrón en la distribución de los residuales que podría estar dada

# por la autocorrelación entre las medidas repetidas (Fecha) en la misma planta.

# Residuales en función de las variables predictoras

```
boxplot(Er~plasti_Bm$Riego, main="Riego", ylab="Residuales estandarizados", xlab="Riego")
```

```
boxplot(Er~plasti_Bm$Sustrato, main="Sustrato", ylab="Residuales estandarizados", xlab="Sustrato")
```

```
plot(Er~plasti_Bm$Fecha, main="Fecha", ylab="Residuales estandarizados", xlab="Fecha")
```

```
abline(0,0, col="red", lw=2)
```

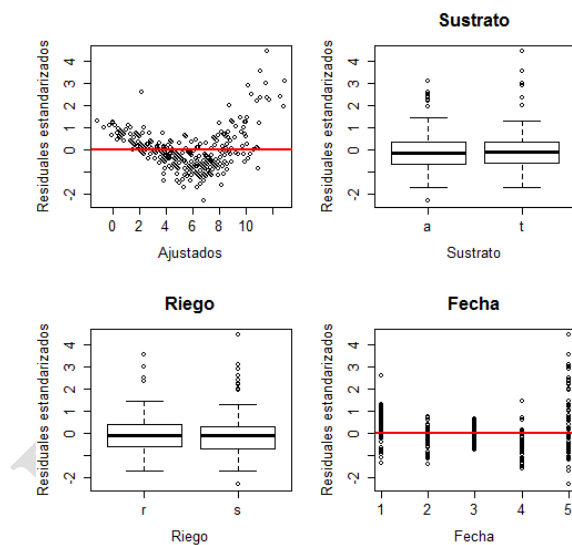


Figura 7. 4: residuales normalizados del modelo "Bm\_mod" en relación a los valores ajustados por el modelo y cada uno de los predictores (Sustrato, riego y fecha).

### # 7- AUTOCORRELACIÓN

#### # 7.a- Evaluar la correlación temporal entre los residuos

```
plot(ACF(Bm_mod, resType= "normalized", na.action=na.omit), alpha=0.05, grid=TRUE)
```

# La "función de auto-correlación" (ACF) indica con una línea punteada el nivel de significancia.



# Las líneas llenas verticales que sobrepasan la línea punteada indican que hay autocorrelación.

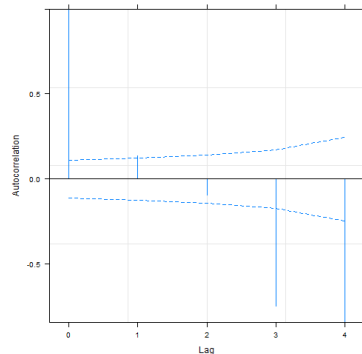


Figura 7. 5: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Bm\_mod". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.

# **¡IMPORTANTE!** Esto sugiere que debemos incluir una función de autocorrelación en el modelo.

# La función de autocorrelación toma valores entre -1 y 1. Al agregarla al modelo # asumimos que existe correlación entre los residuales debida la diferencia de tiempo entre mediciones.

# La función de autocorrelación ACF determina la correlación de cada residual con sí mismo.

# (Lag 0. Observar Lag 0 = 1),

# con el siguiente (Lag 1), con el tercero (Lag 2) y así sucesivamente.

# Dado que en este caso hay 5 mediciones, el máximo Lag es 4.

# **¡IMPORTANTE!** Para calcular las correlaciones de manera correcta es fundamental ordenar los datos siguiendo la estructura de jerarquía de los datos.

# Recordemos que esto lo hicimos antes de ajustar el modelo con la función "order".

### # 7.b- Modelos que incluyen funciones de correlación

# Buscamos la estructura "óptima" de correlación

# Usamos el método REML para comparar distintas estructuras aleatorias.

# Generamos modelos alternativos agregando estructuras de correlación al modelo inicial.

# **¡ATENCIÓN!**: ¡ESTOS MODELOS PUEDEN DEMORAR EN CORRER!

# La siguiente estructura es la autoregresiva de primer orden

```
Bm_modAR1 <- update (Bm_mod, correlation = corAR1(form = ~ Fecha))
```

# Esta estructura es de primer orden para intervalos iguales.

# La forma (form) de la estructura indica que las medidas están repetidas en "Fecha"

```
plot(ACF(Bm_modAR1,resType="normalized"),alpha=0.05,
grid=TRUE,form=~1|fplanta)
```

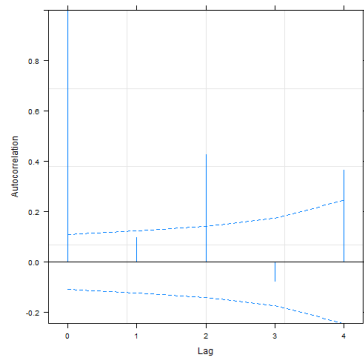


Figura 7. 6: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Bm\_mod". Las líneas llenas veticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.

# Observar que aún hay autocorrelación. En consecuencia,  
# esta estructura no sería la más adecuada para el modelo

# Existen otras estructuras de correlación. Veamos la ayuda sobre corClasses  
?corClasses

# las siguientes estructuras son algunas ARMA

```
Bm_mod10<-update (Bm_mod,correlation=corARMA(p=1,q=0))
plot(ACF(Bm_mod10,resType="normalized"),alpha=0.05, grid=TRUE,
form=~1|fplanta)
```

```
Bm_mod11<-update (Bm_mod,correlation=corARMA(p=1,q=1))
plot(ACF(Bm_mod11,resType="normalized"),alpha=0.05,
grid=TRUE,form=~1|fplanta)
```

```
Bm_mod12<-update (Bm_mod,correlation=corARMA(p=1,q=2))
plot(ACF(Bm_mod12,resType="normalized"),alpha=0.05, grid=TRUE,
form=~1|fplanta)
```

```
Bm_mod20<-update (Bm_mod,correlation=corARMA(p=2,q=0))
plot(ACF(Bm_mod20,resType="normalized"),alpha=0.05, grid=TRUE,
form=~1|fplanta)
```

```
Bm_mod22<-update (Bm_mod,correlation=corARMA(p=2,q=2))
plot(ACF(Bm_mod22,resType="normalized"),alpha=0.05, grid=TRUE,
form=~1|fplanta)
```

## # 8- Comparamos los modelos

# Comparamos los modelos y evaluamos si el ajuste mejora al

# agregar la función de correlación

```
library("bbmle")
```

```
AICTab(Bm_mod, Bm_modAR1, Bm_mod10, Bm_mod11, Bm_mod12, Bm_mod20,
```

```
Bm_mod22, base=T, sort=T, weights=T)
```

	AIC	dAIC	df	weight
Bm_mod12	1289.6	0.0	13	0.73
Bm_mod22	1291.6	2.0	14	0.27
Bm_modAR1	1310.9	21.3	11	<0.001

```
Bm_mod10 1310.9 21.3 11 <0.001
Bm_mod20 1311.4 21.8 12 <0.001
Bm_mod11 1312.1 22.5 12 <0.001
Bm_mod   1376.1 86.5 10 <0.001
```

# El modelo "Bm\_mod12" es el de mejor ajuste entre el conjunto de modelos comparados

### # Verificamos la validez

```
layout(matrix(1:4,2,2))
```

```
# Evaluamos los residuales vs ajustados del mejor modelo
```

```
Er12<-resid(Bm_mod12, type="normalized")
```

```
Fit12<-fitted(Bm_mod12)
```

```
plot(x=Fit12, y=Er12, xlab="Ajustados", ylab="Residuales estandarizados")
```

```
abline(a=0, b=0, col="red", lw=2)
```

# **Observar:** el patrón aun persiste.

```
# Variables predictoras vs. residuos
```

```
boxplot(Er12~plasti_Bm$Sustrato, main="Sustrato", xlab="Sustrato",
ylab="Residuales estandarizados")
```

```
boxplot(Er12~plasti_Bm$Riego, main="Riego", xlab="Riego", ylab="Residuales
estandarizados")
```

```
boxplot(Er12~plasti_Bm$Sustrato:plasti_Bm$Riego, main="Sustrato*Riego",
xlab="Sustrato*Riego", ylab="Residuales estandarizados")
```

# No observamos un patrón de heterogeneidad a nivel de Sustratos \* Riego

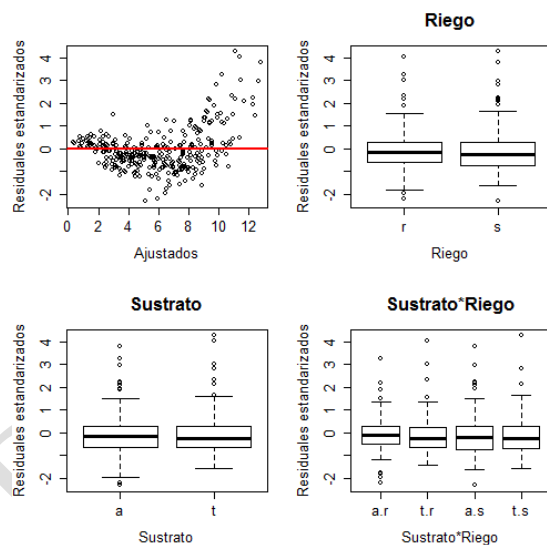


Figura 7. 7: residuales normalizados del modelo "Bm\_mod12" en relación a los valores ajustados por el modelo y cada uno de los predictores (Sustrato, Riego e interacción Sustrato x Riego).

## # 9- Varianzas heterogéneas

### # 9.a- Modelamos las varianzas

# Generamos modelos alternativos que incluyen distintas funciones para modelar las varianzas del factor "Fecha".

```
Bm_mod12.vp<-update (Bm_mod12, weights=varPower(form=~Fecha))
plot(Bm_mod12.vp)
```

```
Bm_mod12.ve<-update (Bm_mod12, weights=varExp(form=~Fecha))
plot(Bm_mod12.ve)
```

```
Bm_mod12.vf<-update (Bm_mod12, weights=varFixed(~Fecha))
plot(Bm_mod12.vf)
```

### # 9.b- Comparamos los modelos

```
AICtab(Bm_mod12, Bm_mod12.vp, Bm_mod12.vf, Bm_mod12.ve, sort=T, base=T,
weights=T)
```

	AIC	dAIC	df	weight
Bm_mod12.ve	1103.7	0.0	14	1
Bm_mod12.vp	1120.5	16.8	14	<0.001
Bm_mod12.vf	1171.2	67.5	13	<0.001
Bm_mod12	1289.6	185.9	13	<0.001

# El modelo que incluye la función de la varianza exponencial "Bm\_mod12.ve" es el de mejor ajuste dentro de este conjunto de modelos.

### # 9.c- Evaluar la presencia de autocorrelación en este nuevo modelo

```
plot(ACF(Bm_mod12.ve,resType="normalized"),alpha=0.05,
grid=TRUE,form=~Fecha|fplanta)
```

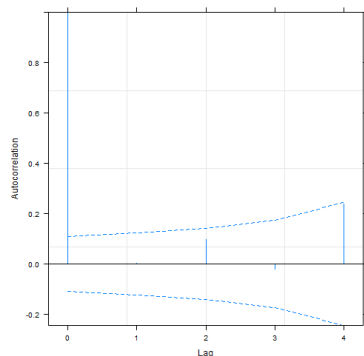


Figura 7. 8: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Bm\_mod12". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.

```
# Evaluamos los residuales vs ajustados del mejo modelo
Er12.ve<-resid(Bm_mod12.ve, type="normalized")
Fit12.ve<-fitted(Bm_mod12.ve)
plot(x=Fit12.ve, y=Er12.ve, xlab="Ajustados", ylab="Residuales estandarizados")
abline(a=0, b=0, col="red", lw=2)
```

# **Observar:** si bien observamos un residual negativo extremo, el patrón en "U" ya no aparece.

# Variables predictoras vs. residuos

```
boxplot(Er12.ve~plasti_Bm$Sustrato, main="Sustrato", xlab="Sustrato",
ylab="Residuales estandarizados")
boxplot(Er12.ve~plasti_Bm$Riego, main="Riego", xlab="Riego",
ylab="Residuales estandarizados")
```

```
boxplot(Er12.ve~plasti_Bm$Sustrato:plasti_Bm$Riego, main="Sustrato*Riego",
xlab="Sustrato*Riego", ylab="Residuales estandarizados")
```

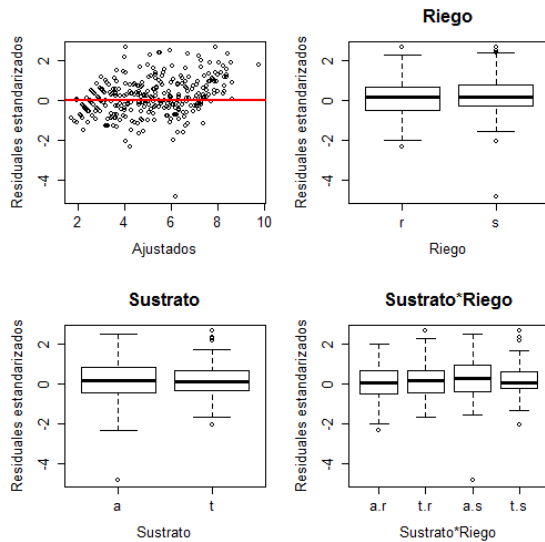


Figura 7. 9: residuales normalizados del modelo "Er12.ve" en relación a los valores ajustados por el modelo y cada uno de los predictores (Sustrato, Riego e interacción Sustrato x Riego).

# 10- Veamos la información que nos da el modelo

```
names(Bm_mod12.ve)
[1] "modelstruct" "dims" "contrasts" "coefficients" "varFix"
[6] "sigma" "apVar" "logLik" "numIter" "groups"
[11] "call" "terms" "method" "fitted" "residuals"
[16] "fixDF" "na.action" "data"
```

# 11- Inferencia sobre los efectos fijos

# Dado que se trata de un experimento manipulativo, utilizaremos como marco inferencial el ANOVA

```
anova(Bm_mod12.ve)
numDF denDF F-value p-value
(Intercept) 1 252 305.98052 <.0001
Riego 1 60 0.74727 0.3908
Sustrato 1 60 2.18047 0.1450
Fecha 1 252 228.00906 <.0001
Riego:Sustrato 1 60 2.12407 0.1502
Riego:Fecha 1 252 1.49435 0.2227
Sustrato:Fecha 1 252 0.52407 0.4698
Riego:Sustrato:Fecha 1 252 0.07138 0.7896
```

# La salida muestra los grados de libertad del numerador (NumDF), los grados de libertad del denominador (denDF).

# El valor del estadístico F (F-value) y el valor de probabilidad asociado (p-value)

# Observar los grados de libertad de los factores

# La altura de las plantas de *Bacharis medullosa* solo dependió del tiempo.

# 12- Gráfico del modelo

```
cl<-with(plasti_Bm, length(levels(interaction(Riego,Sustrato))))
plasti_Bm$trat<-interaction(plasti_Bm$Riego,plasti_Bm$Sustrato)
```

```
layout(matrix(1:1,1,1))
plot(x=plasti_Bm$Fecha, y = plasti_Bm$Altura,
     type = "n",lwd=2 ,col="gray",
     ylab="Altura (cm)", xlab="Feha")
# type="n" indica que el gráfico esta vacío.
# Agregamos los puntos con un color separado por cada tratamiento de
fertilización
```

### ## Riego r + Sustrato a

```
points(plasti_Bm$Fecha[plasti_Bm$trat ==
"r.a"],plasti_Bm$Altura[plasti_Bm$trat == "r.a"],
      col = "black", pch = 16)
fixef(Bm_mod12.ve)
# la función "fixef" muestra los valores estimados por el modelo para los efectos
fijos y la intercepción "promedio ponderada"
ranef(Bm_mod12.ve)
# la función "ranef" muestra los valores estimados para el factor aleatorio. En
este caso, una ordenada al origen para cada planta.
curve(fixef(Bm_mod12.ve)[1]+
      fixef(Bm_mod12.ve)[4]*(x),add=T, col="black", lw=2 )
```

### ## Riego r + Sustrato t

```
points(plasti_Bm$Fecha[plasti_Bm$trat ==
"r.t"],plasti_Bm$Altura[plasti_Bm$trat == "r.t"],
      col = "green", pch = 16)

curve(fixef(Bm_mod12.ve)[1]+
      fixef(Bm_mod12.ve)[3]+
      fixef(Bm_mod12.ve)[4]*(x)+
      fixef(Bm_mod12.ve)[7]*(x),add=T, col="green", lw=2 )
```

### ## Riego s + Sustrato a

```
points(plasti_Bm$Fecha[plasti_Bm$trat ==
"s.a"],plasti_Bm$Altura[plasti_Bm$trat == "s.a"],
      col = "red", pch = 16)

curve(fixef(Bm_mod12.ve)[1]+
      fixef(Bm_mod12.ve)[2]+
      fixef(Bm_mod12.ve)[4]*(x)+
      fixef(Bm_mod12.ve)[6]*(x),add=T, col="red", lw=2 )
```

### ## Riego s + Sustrato t

```
points(plasti_Bm$Fecha[plasti_Bm$trat ==
"s.t"],plasti_Bm$Altura[plasti_Bm$trat == "s.t"],
      col = "blue", pch = 16)

curve(fixef(Bm_mod12.ve)[1]+
      fixef(Bm_mod12.ve)[2]+
```

```

fixef(Bm_mod12.ve)[3]+
fixef(Bm_mod12.ve)[5]+
fixef(Bm_mod12.ve)[4]*(x)+
fixef(Bm_mod12.ve)[6]*(x)+
fixef(Bm_mod12.ve)[7]*(x)+
fixef(Bm_mod12.ve)[8]*(x),add=T, col="blue", lw=2 )

```

```

legend("topleft",levels(plasti_Bm$strat), col=1:cl, pch=16, bty="n")

```

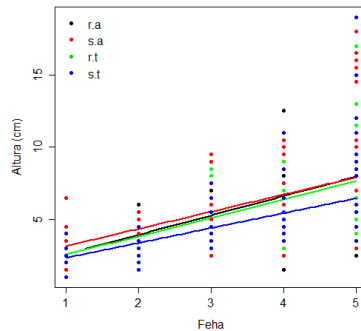


Figura 7. 10: altura de las plantas de *Baccharis medullosa* (cm) en el tiempo (fechas) y ajuste general del modelo (líneas llenas). Los colores indican los tratamientos: r.a (riego y arena, negro), s.a (sin riego y arena, rojo), r.t (riego y tierra, verde), s.t (sin riego y tierra, azul).

### # 13- Resumen

# Escribimos el modelo de efectos mixtos (o de "partial pooling") que contempla la estructura dada por el diseño del experimento y el método de muestreo.

# Observamos el patrón en los residuales del modelo.

# Evaluamos la correlación dada por las medidas repetidas en el tiempo mediante la "función de auto-correlación" (ACF).

# Encontramos la estructura óptima para los componentes aleatorios mediante funciones de correlación (autorregresivas) y, funciones de la varianza.

# Evaluamos el componente fijo del modelo en este caso mediante un análisis de la deviance (método ML).

# Finalmente, presentamos el modelo utilizando el método REML de estimación.

### # 14- Tarea

# a- ¿Cómo hubiera sido el modelo si, en lugar del modelo lineal ajustamos un modelo con el  $\log(\text{Altura})$  como predictor?

# Repita el procedimiento utilizando  $\log(\text{Altura})$  como predictor y explique las diferencias observadas.

# b- Repetir el/los procedimientos analizando la altura de la otra especie, *E. buniifolium*.

### # 15- Agradecimiento

# Datos provistos por Fernando Biganzoli y William B. Batista de la Facultad de Agronomía de la Universidad de Buenos Aires. Biganzoli, F., E. Fernández, R. J. Fernández & W. B. Batista. "Diferencias en el uso del agua en plántulas de dos especies arbustivas de la sabana del Parque Nacional El Palmar, Entre Ríos". Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.

# Autocorrelación – Medidas repetidas con intervalos iguales II

## Contenidos

### # CAPÍTULO 8: Autocorrelación – Medidas repetidas con intervalos iguales

<b>II</b>	<b>121</b>
# 1- Configuración inicial	121
# 1.a- Definir el directorio de trabajo	121
# 1.b- Leer el archivo de datos	121
# 2- El caso: Determinantes de la calidad maltera de cebada cervecera	121
# Diseño	122
# 2.a- Pregunta de interés	122
# 2.b- Variable respuesta	123
# 2.c- Variable/s predictoras de interés	123
# 3- Identificar jerarquías	123
# 4- Explorar los datos	123
# 4.a- Outliers	123
# 4.b- Multicolinealidad	125
# 5- Construcción del modelo inicial	126
# 5.a- Modelo inicial lineal	126
# 5.c- Modelo cuadrático	128
# 5.d- Comparaciones gráficas de los modelos	129
# 5.e- Verificamos la validez del Modelo cuadrático	130
# 6- Comparamos los modelos	131
# 7- Ajustamos la componente aleatoria del modelo	131
# 7.a- Correlación temporal	132
# 7.b- Agregamos estructuras de autocorrelación al modelo	133
# 7.c- Verificación de la validez del modelo	134
# 7.d- Funciones de la varianza	135
# 8- Inferencia sobre los efectos fijos	138
# 8.a- Generamos múltiples modelos	138
# 8.b- Ordenar los modelos según su bondad de ajuste	139
# 8.c- Identificar el mejor modelo	141
# 8.d- Identificar el modelo nulo	142
# 8.e- Verificar el mejor modelo	142
# 9- Gráfico del modelo	144
# 10- Resumen	145
# 11- Tarea	145
# 12- Agradecimiento	146



## # CAPÍTULO 8: Autocorrelación – Medidas repetidas con intervalos iguales II

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo de datos

```
# horde<-.....(Buscar ayuda y la tabla de datos "8_Hordeina.csv")
summary(horde)
```

# Tenemos una tabla de datos con algunos datos faltantes (NA's).

# Dado que esta variable tiene datos faltantes, generamos una tabla que los excluye.

```
horde2<-subset(horde,Hord_B!="NA")
```

```
View(horde2)
```

# Este paso, facilita luego generar los gráficos de ajuste de los modelos.

```
with(horde2, table(Parcela))
```

# La tabla muestra que el set de datos posee entre 3 y 6 medidas repetidas por parcela.

```
with(horde2, table(dpa))
```

```
dpa
```

```
 8 15 22 29 36 43
25 56 58 60 60 28
```

# También vemos que tenemos entre 25 y 60 datos por fecha.

### # 2- El caso: Determinantes de la calidad maltera de cebada cervecera



Las hordeínas son proteínas características del grano de cebada que contienen nitrógeno y azufre. El grano de cebada produce hordeínas B, C, D y gamma. La cantidad de estas hordeínas en el grano influye sobre la calidad maltera para producción de cerveza. Sin embargo, aun se sabe poco respecto de la producción de cada una de estas hordeínas a lo largo de la ontogenia (o formación) del grano, el impacto de la fertilización y, su relación con la calidad maltera de la cebada. Para estudiar estos aspectos se realizó un experimento bajo condiciones ambientales naturales (Fig.

8.1). En 4 localidades de la región pampeana (Junin, Baigorrita, Tiburcio y Campito) se instalaron 16 parcelas (Total, 64 parcelas). En cada localidad se aplicaron tratamientos de fertilización según el arreglo factorial de los factores fertilización con nitrógeno y fertilización con azufre (2 niveles cada uno. En total 4 repeticiones de los 4 tratamientos por localidad). A lo largo de la ontogenia del grano, es decir, en fechas sucesivas después de anthesis se tomaron muestras de 10 espigas de cebada por parcela. Las muestras fueron utilizadas para determinar el número de granos, el peso de mil semillas (g), la cantidad de hordeínas B, C, D y Gamma (Las determinaciones de hordeínas se realizaron un HPLC y el valor en la tabla de datos corresponde al área determinada por el equipo).

Este experimento lo utilizaremos para evaluar: i) el efecto de la fertilización sobre la producción de hordeínas B y, ii) el efecto de la fertilización sobre la producción de hordeínas B durante la ontogenia del grano.

iii) Para evaluar la influencia del nitrógeno y el azufre en la producción de hordeínas B en grano de cebada utilizaremos los valores de hordeínas B a los 36 días después de antesis. En este momento se considera que el grano ha alcanzado el máximo nivel de hordeínas. Las variables predictoras serán: i) la fertilización con nitrógeno (N, dos niveles), ii) la fertilización con azufre (S, dos niveles) y, iii) el peso de mil semillas (g). La estructura de dependencia está dada por las parcelas en los ambientes.

iv) Para evaluar el efecto de la fertilización a lo largo de la ontogenia del grano, a las variables predictoras anteriores el tiempo. La variable tiempo contempla de seis a siete mediciones por parcela (Se realizaron seis mediciones en tres localidades y siete en la cuarta. Si bien la cantidad de mediciones es la misma, el día post antesis (dpa) no siempre fue el mismo entre localidades).

## # Diseño

### Diseño del experimento

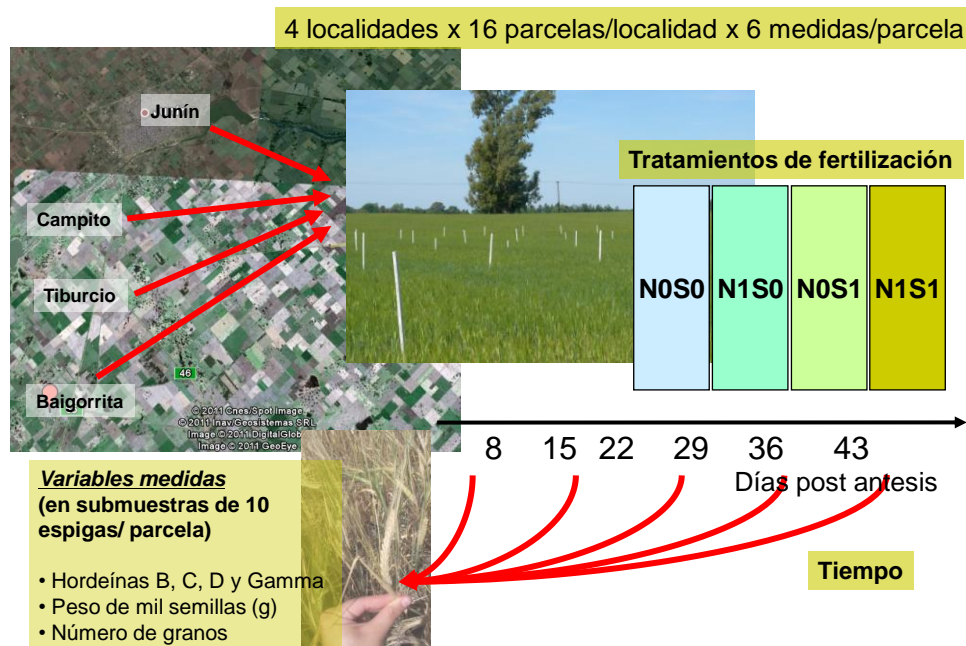


Figura 8. 1: diseño del experimento

### # 2.a- Pregunta de interés

# ¿Qué influencia tienen el Nitrógeno y el Azufre en la producción de Hordeínas en grano de cebada?

# Las hordeínas son proteínas características del grano de cebada que contienen nitrógeno y azufre.

# El grano de cebada produce hordeínas B, C, D y gamma. La cantidad de estas hordeínas en el grano influye sobre la calidad maltera para producción de cerveza.

# Sin embargo, aun se sabe poco respecto de la producción de cada una de estas hordeínas a lo largo de la ontogenia (o formación) del grano.

**# 2.b- Variable respuesta**

# Cantidad de hordeínas B (Si bien se cuantificaron los cuatro tipos de hordeínas desarrollamos el ejemplo con un solo tipo).

# La cuantificación de hordeínas no posee unidades. Es el área bajo la curva que determina el equipo (HPLC) utilizado para hacer la determinación.

# La cuantificación de hordeínas se realizó a partir de una submuestra de "10" espigas por fecha y parcela.

**# 2.c- Variable/s predictoras de interés**

# Fecha [días post antesis (dpa)]

# Fertilización: factorial 2 factores (N y S) con 2 niveles cada uno

# El peso de mil semillas (g)

**# 3- Identificar jerarquías**

# ¿Cuáles son las escalas y qué variables se midieron en cada uno?

# Fechas (6 a 7) por parcela: cantidad de hordeínas B en seis o siete fechas post antesis (dpa)

# peso de mil semillas (granos, PMS[g])

# Se realizaron seis mediciones en tres localidad y siete en la cuarta.

# Si bien la cantidad de mediciones es la misma, el día post antesis (dpa) no siempre fue el mismo entre localidades.

# Parcelas (16) por localidad: donde se aplicaron 4 tratamientos en un arreglo factorial de fertilización (N x S)

# Localidades (4): Junin, Baigorrita, Campito, Tiburcio

# Es decir, Fechas en Parcelas en Localidades

**# 4- Explorar los datos****# 4.a- Outliers****# 4.a.1- Gráficos básicos**

```
with(horde2, plot(N, Hord_B, xlab="Fertilización con N", ylab="Hordeínas B"))
```

```
with(horde2, plot(S, Hord_B, xlab="Fertilización con S", ylab="Hordeínas B"))
```

```
with(horde2, plot(N:S, Hord_B, xlab="Fertilización con S y N", ylab="Hordeínas B", las=3))
```

```
with(horde2, plot(Localidad, Hord_B, xlab="Localidad", ylab="Hordeínas B"))
```

```
with(horde2, plot(PMS, Hord_B, xlab="Peso de mil semillas (g)", ylab="Hordeínas B"))
```

```
with(horde2, plot(dpa, Hord_B, xlab="Días post Antesis", ylab="Hordeínas B"))
```

# En particular, este último gráfico sugiere que puede haber una relación no lineal entre Hord\_B y dpa.

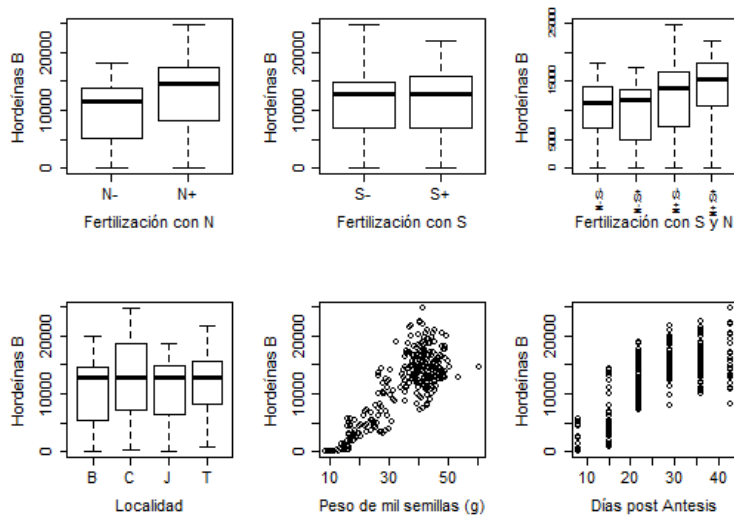


Figura 8. 2: exploración de los datos. Valores de hordeínas B en relación a los predictores (de izquierda superior a derecha inferior): niveles de fertilización con nitrógeno, niveles de fertilización con azufre, la interacción nitrógeno x azufre, las localidades, el peso de mil semillas y los días post antesis de *Hordeum vulgare*.

#### # 4.a.2- Gráficos distribución de puntos (lattice)

```
library("lattice")
```

```
with(horde2, xyplot(Hord_B ~ dpa|Localidad, main="Hordeínas B post antesis  
por Localidad", xlab="Días post Antesis", ylab="Hordeínas B"))
```

```
with(horde2, xyplot(Hord_B ~ dpa|N, main="Hordeínas B post antesis según fert.  
con N", xlab="Días post Antesis", ylab="Hordeínas B"))
```

```
with(horde2, xyplot(Hord_B ~ dpa|S, main="Hordeínas B post antesis según fert.  
con S", xlab="Días post Antesis", ylab="Hordeínas B"))
```

```
with(horde2, xyplot(Hord_B ~ dpa|S*N, main="Hordeínas B post antesis según  
fert. con N y S", xlab="Días post Antesis", ylab="Hordeínas B"))
```

# Este tipo de gráficos tiene la virtud de combinar tres variables. Es útil para explorar por ejemplo si la relación entre hordeínas B y los días post antesis cambia entre localidades o tratamientos de fertilización.

# Este tipo de gráficos también puede ser útil para evaluar la distribución de residuales de un modelo.

# Podemos realizar gráficos similares para el peso de mil semillas

#### # 4.a.3- Gráficos distribución de puntos II

```
with(horde2, dotchart(Hord_B, ylab="Conjunto de datos", xlab="Hor deínas B"))
```

```
with(horde2, dotchart(Hord_B, groups=Localidad, ylab="Localidad",  
xlab="Hordeínas B"))
```

```
with(horde2, dotchart(Hord_B, groups=N, ylab="Fertilización con N",  
xlab="Hordeínas B"))
```

```
with(horde2, dotchart(Hord_B, groups=S, ylab="Fertilización con S",  
xlab="Hordeínas B"))
```

```
with(horde2, dotchart(Hord_B, groups=interaction(S,N), ylab="Datos según  
fertilización", xlab="Hordeínas B"))
```

# Este tipo de gráfico también nos ayuda a observar diferencias en los valores y en la variabilidad de los datos entre grupos.

# COMPLETAR LAS SENTENCIAS usando como guía el CAPÍTULO 2

#### # 4.b- Multicolinealidad

# Evaluamos multicolinealidad: "PMS", "dpa" ("Localidad", "N" y "S" quedan excluidas por ser categóricas).

# **IMPORTANTE:** la multicolinealidad la evaluamos entre variables predictoras.

```
panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, use="complete.obs"))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}
```

```
pairs(horde2[,c("PMS", "dpa")],
      lower.panel = panel.cor,
      upper.panel= panel.smooth,
      cex.labels=1,
      labels=c("PMS_gr", "dpa"))
```

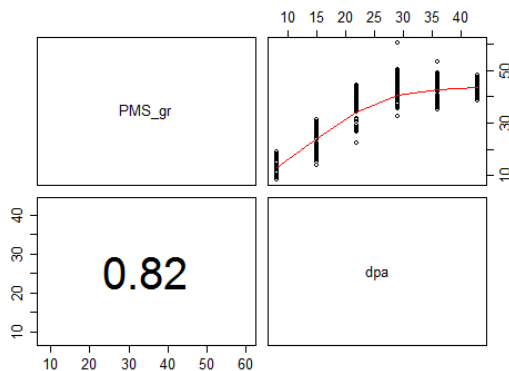


Figura 8. 3: relación entre variable cuantitativas: peso de mil semillas (g) y días post antesis (dpa) de *Hordeum vulgare*.

# Una forma de evaluar este tipo de colinealidad es comparar el  $R^2$  de funciones lineales donde cada una de las predictoras funciona como variable respuesta.

```
col1<-lm(PMS~dpa*N*S, data=horde2)
cor(horde2$PMS, fitted(col1))^2
[1] 0.6841255
```

```
col2<-lm(dpa~PMS*N*S, data=horde2)
cor(horde2$dpa, fitted(col2))^2
[1] 0.6833567
```

# El ajuste similar entre ambos modelos está relacionado a la alta correlación que existe entre ambas.

# La variable predictora que genera un alto  $R^2$  presenta colinealidad con las demás variables predictoras del modelo y debería ser excluida.

```
1/(1-cor(horde2$dpa, fitted(col2))^2)
```

```
[1] 3.158128
```

# **IMPORTANTE:** Existe multicolinealidad (alta correlación) entre días post anthesis (dpa) y el peso de mil semillas (PMS). Por lo tanto, no podremos incluir a ambas variables como predictoras en nuestro modelo.

# Utilizaremos: dpa (dado que nos interesa especialmente la ontogenia del grano)

### # 5- Construcción del modelo inicial

# Modelo inicial de varianzas homogéneas.

# **ATENCIÓN:** es importante ordenar los datos siguiendo la estructura de jerarquías.

# Esto se debe a que la función que estima la autocorrelación entre los datos es empírica. Es decir, depende del orden en que se encuentran los datos.

```
horde2_ord<-horde2
```

```
# copiamos el mismo set de datos en un nuevo objeto
```

```
ord<-with(horde2_ord, order(Parcela, dpa, Localidad))
```

```
horde2<-horde2_ord[ord,]
```

```
# Ahora podemos ver los datos para verificar el orden
```

```
View(horde2)
```

### # 5.a- Modelo inicial lineal

# Incluimos todas las variables e interacciones sin incluir por ahora una función de correlación para las medidas en el tiempo.

# A partir de este modelo evaluamos si corresponde o no corregir la autocorrelación temporal.

```
library("nlme")
```

```
mod<-lme(Hord_B~N*S*dpa,
         random=~1|Localidad/Parcela,
         data=horde2)
```

# na.action=na.omit, omite las filas de datos faltantes para las variables incluidas en el modelo.

# Si no hubiéramos generado la horde2, los argumentos del modelo hubieran sido: .....data= horde, na.action=na.omit)

# Tener la tabla horde2 facilita los gráficos a continuación

### # 5.a.1- Verificación de la validez del modelo

# Ajustados vs. los residuales del modelo

```
layout(matrix(1:6,2,3,byrow = TRUE))
```

```
Er<-resid(mod, type="normalized")
```

```
Fit<-fitted(mod)
```

```
plot(x=Fit, y=Er, xlab="Ajustados", ylab="Residuales estandarizados")
```

```
abline(a=0, b=0, col="red", lw=2)
```

# **IMPORTANTE:** observar el patrón no lineal en este gráfico.



# Los residuales negativos se encuentran en los extremos y los positivos en el centro de los valores ajustados.

# Debemos plantear un modelo fijo que corrija este patrón.

```
# Residuales en función de las variables predictoras
boxplot(Er~horde2$Localidad, main="Localidad", ylab="Residuales
estandarizados", xlab="Localidad")
boxplot(Er~horde2$Parcela, main="Parcela", ylab="Residuales estandarizados",
xlab="Parcelas")
boxplot(Er~horde2$N, main="N", ylab="Residuales estandarizados",
xlab="Fertilización")
boxplot(Er~horde2$S, main="S", ylab="Residuales estandarizados",
xlab="Fertilización")
plot(Er~horde2$dpa, main="Dias post antesis", ylab="Residuales
estandarizados", xlab="Días post antesis")
abline(0,0, col="red")
# La distribución de los residuales en el último gráfico tiene un patrón similar al
del modelo general y sugiere una relación no lineal entre Hord_B y dpa.
# Esto es coherente con el gráfico exploratorio de la línea 42.
```

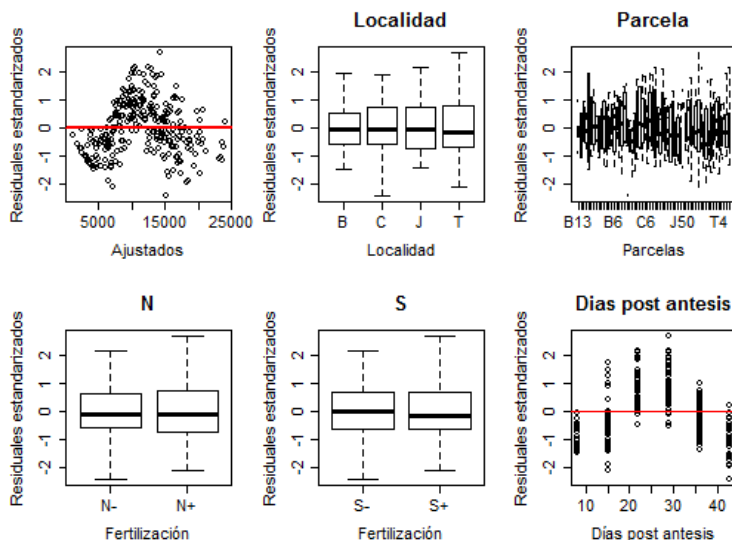


Figura 8. 4: residuales normalizados del modelo "mod" en relación a los valores ajustados por el modelo y cada uno de los predictores (Localidad, Parcela, Fertilización con nitrógeno, Fertilización con azufre y días post antesis. De izquierda a derecha).

# **IMPORTANTE:** debemos cambiar la parte fija del modelo antes de continuar.

# 5.b- Modelo inicial log

# (no lineal)

# Planteamos el modelo con la variable predictora dpa transformada según

log(dpa)

```
mod2<-lme(Hord_B~N*S*log(dpa),
random=~1|Localidad/Parcela,
data=horde2)
```

**# 5.b.1- Verificación de la validez del modelo**

```
# Ajustados vs. los residuales del modelo
layout(matrix(1:6,2,3,byrow = TRUE))
Er2<-resid(mod2, type="normalized")
Fit2<-fitted(mod2)
plot(x=Fit2, y=Er2, xlab="ajustados", ylab="residuales estandarizados")
abline(a=0, b=0, col="red", lw=2)
```

# El patrón en "U" invertida es un poco menos marcado.  
# Además, aparecen valores predichos negativos (izquierda del gráfico).

```
# Residuales en función de las variables predictoras
boxplot(Er2~horde2$Localidad, main="Localidad", ylab="Residuales
estandarizados", xlab="Localidad")
boxplot(Er2~horde2$Parcela, main="Parcela", ylab="Residuales
estandarizados", xlab="Parcelas")
boxplot(Er2~horde2$N, main="N", ylab="Residuales estandarizados",
xlab="Fertilización")
boxplot(Er2~horde2$S, main="S", ylab="Residuales estandarizados",
xlab="Fertilización")
plot(Er2~horde2$dpa, main="Dias post antesis", ylab="Residuales
estandarizados", xlab="Días post antesis")
abline(0,0, col="red")
```

# El modelo con log(dpa) no resolvió el problema de la distribución de los residuos.

**# 5.c- Modelo cuadrático****# ("dpa + dpa<sup>2</sup>", no lineal)**

```
# Planteamos un modelo que incluye "dpa + dpa2" en el componente fijo.
# I(dpa2) es la sintaxis para indicar el término elevado al cuadrado.
mod3<-lme(Hord_B~N*S*dpa*I(dpa2),
random=~1|Localidad/Parcela,
data=horde2)
```

**# 5.c.1- Verificación de la validez del modelo**

```
# Ajustados vs. los residuales del modelo
layout(matrix(1:1,1,1))

Er3<-resid(mod3, type="normalized")
Fit3<-fitted(mod3)
plot(x=Fit3, y=Er3, xlab="ajustados", ylab="residuales estandarizados")
abline(a=0, b=0, col="red", lw=2)
```

# Observamos un patrón en "U" invertida atenuado respecto de los modelos "mod" y "mod2".

# Este modelo también predice valores (hordeinas B) negativos.



## # 5.d- Comparaciones gráficas de los modelos

## # 5.d.1- Residuales vs. predichos

```
layout(matrix(1:3,1,3, byrow=TRUE))
plot(Er~Fit, main="Modelo lineal", cex.main=0.8, ylab="Residuales
normalizados", xlab="Predichos")
abline(0,0, col="red")
abline(0,1, col="red")
```

```
plot(Er2~Fit2, main="Modelo log(dpa)", cex.main=0.8, ylab="",
xlab="Predichos")
abline(0,0, col="red")
abline(0,1, col="red")
```

```
plot(Er3~Fit3, main="Modelo dpa+dpa^2", cex.main=0.8, ylab="",
xlab="Predichos")
abline(0,0, col="red")
abline(0,1, col="red")
```

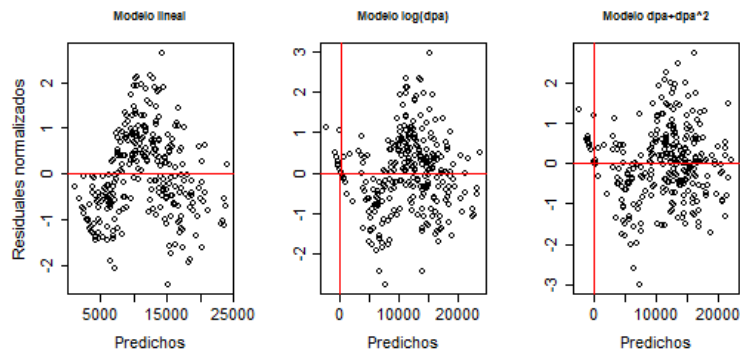


Figura 8. 5: residuales normalizados en relación a los valores ajustados para los modelos lineal (mod. Izq.), logarítmico [ $\log(dpa)$ . mod2. Centro] y, cuadrático [ $dpa+dpa^2$ . mod3. Der.]. La líneas rojas indican los valores de cero para los residuales (horizontal) y los predichos (vertical).

## # 5.d.2- Residuales vs. dpa

```
layout(matrix(1:3,1,3, byrow=TRUE))
plot(Er~horde2$dpa, main="Modelo lineal", ylab="Residuales estandarizados",
xlab="Días post antesis")
abline(0,0, col="red")
plot(Er2~horde2$dpa, main="Modelo log(dpa)", ylab="Residuales
estandarizados", xlab="Días post antesis")
abline(0,0, col="red")
plot(Er3~horde2$dpa, main="Modelo dpa+dpa^2", ylab="Residuales
estandarizados", xlab="Días post antesis")
abline(0,0, col="red")
```

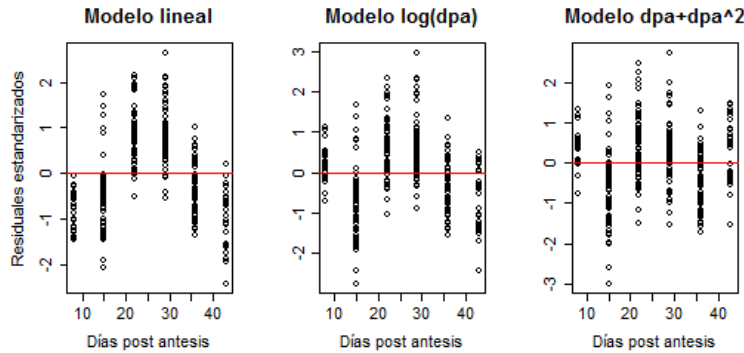


Figura 8. 6: residuales normalizados en relación a los días post antesis (dpa) para los modelos lineal (mod. Izq.), logarítmico [ $\log(\text{dpa})$ . mod2. Centro] y, cuadrático [ $\text{dpa}+\text{dpa}^2$ . mod3. Der.].

### # 5.d.3- Normalidad en la distribución de los residuos

```
layout(matrix(1:3,1,3, byrow=TRUE))
qqnorm(Er, main="Modelo lineal", ylim=c(-3,3), xlim=c(-3,3))
qqline(Er)
qqnorm(Er2, main="Modelo log(dpa)", ylim=c(-3,3), xlim=c(-3,3))
qqline(Er2)
qqnorm(Er3, main="Modelo dpa+dpa^2", ylim=c(-3,3), xlim=c(-3,3))
qqline(Er3)
```

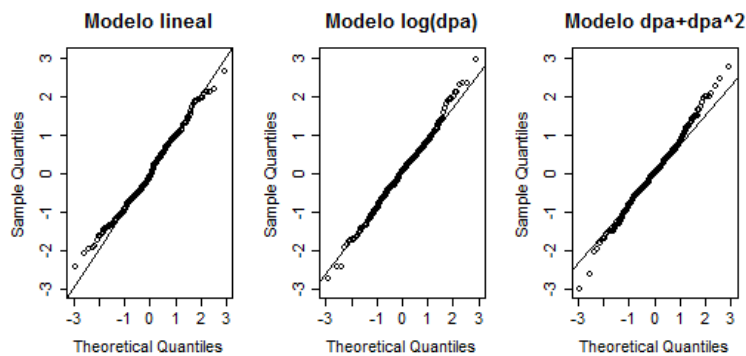


Figura 8. 7: gráfico de normalidad (cuantiles teóricos) para los modelos lineal (mod. Izq.), logarítmico [ $\log(\text{dpa})$ . mod2. Centro] y, cuadrático [ $\text{dpa}+\text{dpa}^2$ . mod3. Der.].

### # 5.e- Verificamos la validez del Modelo cuadrático

# Residuales vs. las demás variables predictoras

```
layout(matrix(1:6,2,3, byrow=TRUE))
boxplot(Er3~horde2$Localidad, main="Localidad", ylab="Residuales estandarizados", xlab="Localidad")
boxplot(Er3~horde2$N, main="N", ylab="Residuales estandarizados", xlab="Fertilización")
boxplot(Er3~horde2$S, main="S", ylab="Residuales estandarizados", xlab="Fertilización")
boxplot(Er3~horde2$N:horde2$S, main="N*S", ylab="Residuales estandarizados", xlab="Fertilización")
plot(Er3~horde2$dpa, main="Días post antesis", ylab="Residuales estandarizados", xlab="Días post antesis")
abline(0,0, col="red")
```

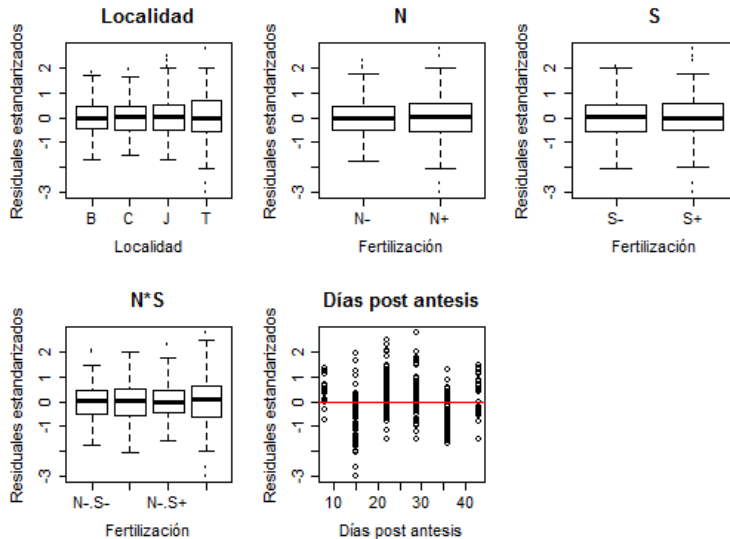


Figura 8. 8: residuales normalizados del modelo "mod3" en relación a los predictores: Localidad, Fertilización con nitrógeno, Fertilización con azufre, la interacción N\*S y días post antesis. De izquierda a derecha.

# Observamos varianzas heterogéneas a nivel de Localidad, Tratamientos de fertilización (N\*S)

# **IMPORTANTE:** hasta aquí probamos distintas estructuras determinísticas para definir el modelo inicial global.

# Una vez definido el modelo global, ajustamos el componente aleatorio para este modelo global.

# **IMPORTANTE:** a pesar que el modelo "mod3" mejora la distribución de los errores, predice valores negativos.

# Continuo la práctica con el modelo "mod" que NO predice valores negativos

### # 6- Comparamos los modelos

# **IMPORTANTE:** para comparar el ajuste de los modelos mediante AIC debemos ajustarlos por el método de ML

# Esto es así porque no son modelos estrictamente anidados unos en otros.

```
modML<-update(mod, method="ML")
```

```
mod2ML<-update(mod2, method="ML")
```

```
mod3ML<-update(mod3, method="ML")
```

```
library("bbmle")
```

```
AICtab(modML, mod2ML, mod3ML, weights=T, base=T, sort=T)
```

```
      AIC    dAIC  df weight
```

```
mod3ML 5284.8     0.0  19  1
```

```
mod2ML 5334.4    49.7  11 <0.001
```

```
modML  5441.4   156.6  11 <0.001
```

# El modelo "mod3ML" tuvo un ajuste sensiblemente mejor que los anteriores.

### # 7- Ajustamos la componente aleatoria del modelo

# **IMPORTANTE:** el orden en que incorporamos funciones en nuestro modelo no es causal (Zuur et al. 2009).

# 1. Factores aleatorios, 2. Heterogeneidad, 3. Correlación temporal, 4. correlación espacial

# Para esta parte volvemos a nuestro mejor modelo ajustado por el método de "REML"

### # 7.a- Correlación temporal

#### # 7.a.1- Evaluar autocorrelación entre los residuos

# Realizamos un variograma para evaluar autocorrelación

```
library("nlme")
```

```
plot(ACF(mod3,resType="normalized",na.action=na.omit),alpha=0.05,
grid=TRUE)
```

# Indicamos la línea de corte al 5%. Lo definimos con el argumento "alpha".

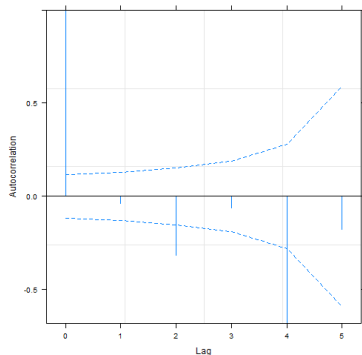


Figura 8. 9: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "mod3". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.

# Las líneas llenas verticales que sobrepasan la línea punteada indican que hay autocorrelación.

# Esto nos sugiere que debemos incluir una función de autocorrelación en el modelo.

# La función de autocorrelación toma valores entre -1 y 1. Al agregarla al modelo asumimos que existe correlación entre los residuales debida la diferencia de tiempo entre mediciones.

# La ACF determina la correlación de cada residual con si mismo (Lag 0.

Observamos Lag 0 = 1), con el siguiente (Lag 1), con el tercero (Lag 2) y así sucesivamente.

# Dado que en este caso hay como máximo 6 mediciones por parcela, el máximo Lag es 5.

# Para que las correlaciones se calculen de manera correcta es fundamental ordenar los datos siguiendo la estructura jerárquica Localidad, Parcela, dpa.

# Recordemos que esto lo hicimos antes de ajustar los modelos (5 - Construcción del modelo inicial. Línea 113).

# La "función de auto-correlación" (ACF) indica con una línea punteada el nivel de significancia.

# Las líneas llenas verticales que sobrepasan la línea punteada indican que hay autorrelación.

# ¡**IMPORTANTE!** Esto sugiere que debemos incluir una función de autocorrelación en el modelo.

### # 7.b- Agregamos estructuras de autocorrelación al modelo

# Planteamos modelos con otras estructuras de correlación para mejorar el ajuste.

# ¿Cuántas estructuras existen? veamos en help: corClasses  
?corClasses

# La siguiente estructura es autorregresiva de primer orden (AR1)

```
mod.AR1<-update(mod3, correlation=corAR1(form=~1|Localidad/Parcela))
summary(mod.AR1)
```

```
Correlation Structure: AR(1)
Formula: ~1 | Localidad/Parcela
Parameter estimate(s):
  Phi
0.3107914
```

```
plot(ACF(mod.AR1, resType="normalized", na.action=na.omit), alpha=0.05,
grid=TRUE)
```

# Las siguientes estructuras son ARMA (autorregresivas con media móvil)

```
mod.ARMA10<-update (mod3, correlation=corARMA(p=1,q=0,
form=~1|Localidad/Parcela))
```

```
summary(mod.ARMA10)
Correlation Structure: AR(1)
Formula: ~1 | Localidad/Parcela
Parameter estimate(s):
  Phi
0.3107914
```

```
plot(ACF(mod.ARMA10, resType="normalized", na.action=na.omit), alpha=0.05,
grid=TRUE)
```

# Observar: los valores de Phi estimado en el modelo con la estructura corAR1 y ARMA (1,0) son equivalentes

```
mod.ARMA20<-update (mod3, correlation=corARMA(p=2,q=0,
form=~1|Localidad/Parcela))
```

```
summary(mod.ARMA20)
Correlation Structure: ARMA(2,0)
Formula: ~1 | Localidad/Parcela
Parameter estimate(s):
  Phi1      Phi2
0.27526104 -0.06357973
```

# Observar que el modelo estima dos parámetros: Phi1 y Phi2

```
plot(ACF(mod.ARMA20, resType="normalized", na.action=na.omit), alpha=0.05,
grid=TRUE)
```

```
mod.ARMA30<-update (mod3, correlation=corARMA(p=3,q=0,
form=~1|Localidad/Parcela))
```

```
summary(mod.ARMA30)
Correlation Structure: ARMA(3,0)
Formula: ~1 | Localidad/Parcela
Parameter estimate(s):
```

```

      Phi1      Phi2      Phi3
0.4050575 -0.1111150  0.4259476
plot(ACF(mod.ARMA30, resType="normalized", na.action=na.omit), alpha=0.05,
grid=TRUE)

```

```

mod.ARMA11<-update (mod3, correlation=corARMA(p=1,q=1,
form=~1|Localidad/Parcela))

```

```

summary(mod.ARMA11)
Correlation Structure: ARMA(1,1)
Formula: ~1 | Localidad/Parcela
Parameter estimate(s):
  Phi1      Theta1
-0.5769438  0.9998559

```

```

# Observar que ahora el modelo estima dos parámetros: Phi1 y Theta1
plot(ACF(mod.ARMA11, resType="normalized", na.action=na.omit), alpha=0.05,
grid=TRUE)

```

# Otras estructuras de correlación las utilizaremos para casos de correlación especial.

```

# Ej.: corExp(form=~....), corGaus(form=~....), corLin(form=~....)

```

```

# Para todos los modelos podemos hacer un gráfico como los anteriores
# plot(ACF(mod....))

```

# Comparamos los dos modelos y evaluamos si el ajuste mejora significativamente al agregar la correlación.

```

library("bbmle")

```

```

AICtab(mod3, mod.AR1, mod.ARMA10, mod.ARMA20, mod.ARMA30,
mod.ARMA11, weights = T, delta = T, base = T, sort = T)

```

	AIC	dAIC	df	weight
mod.ARMA30	5171.7	0.0	22	0.513
mod.ARMA11	5172.1	0.4	21	0.426
mod.AR1	5177.7	6.0	20	0.025
mod.ARMA10	5177.7	6.0	20	0.025
mod.ARMA20	5179.5	7.9	21	0.010
mod3	5184.9	13.2	19	<0.001

# El modelo con la estructura ARMA de tercer orden: "mod.ARMA30" tiene la mejor bondad de ajuste entre los modelos comparados.

# Además, el gráfico de autocorrelación no muestra correlaciones importantes.

# Observar: el modelo con la estructura ARMA (1,1): "mod.ARMA11" tiene una bondad de ajuste similar.

# Sin embargo, el gráfico de autocorrelación muestra correlación entre mediciones.

### # 7.c- Verificación de la validez del modelo

#### # 7.c.1- Ajustados vs. los residuales del modelo

```

layout(matrix(1:6,2,3,byrow = TRUE))

```

```

Ercor<-resid(mod.ARMA30, type="normalized")

```

```

Fitcor<-fitted(mod.ARMA30)

```

```

plot(x=Fitcor, y=Ercor, xlab="ajustados", ylab="residuales estandarizados")

```

```

abline(a=0, b=0, col="red", lw=2)

```

```

abline(a=0, b=1, col="red", lw=2)

```

# El patrón en "U" invertida es muy poco marcado.  
 # Sin embargo, aparecen algunos predichos negativos (izquierda del gráfico).

# 7.c.2- Residuales en función de las variables predictoras

```

boxplot(Ercor~horde2$Localidad, main="Localidad", ylab="Residuales
estandarizados", xlab="Localidad")
boxplot(Ercor~horde2$Parcela, main="Parcela", ylab="Residuales
estandarizados", xlab="Parcelas")
boxplot(Ercor~horde2$N, main="N", ylab="Residuales estandarizados",
xlab="Fertilización")
boxplot(Ercor~horde2$S, main="S", ylab="Residuales estandarizados",
xlab="Fertilización")
plot(Ercor~horde2$dpa, main="Dias post antesis", ylab="Residuales
estandarizados", xlab="Días post antesis")
abline(a=0,b=0, col="red")
    
```

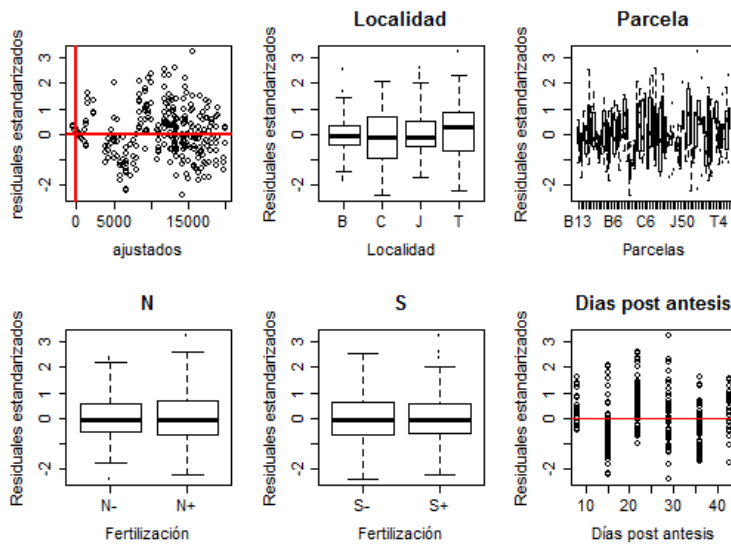


Figura 8. 10: residuales normalizados del modelo "mod.ARMA30" en relación a los valores ajustados por el modelo y cada uno de los predictores: Localidad, Parcela, Fertilización con nitrógeno, Fertilización con azufre y días post antesis. De izquierda a derecha. Las líneas rojas indican valores de cero para los residuales (horizontal) y los ajustados (vertical).

# 7.d- Funciones de la varianza

# Incorporamos la función (varIdent) que pondera una varianza diferente para cada nivel del factor.  
 # Recordemos que esta función incorpora nuevos parámetros al modelo (tantos como niveles del factor-1)

# Localidad

```
mod4<-update(mod.ARMA30, weights = varIdent(form=~ 1 | Localidad))
```

# Nitrógeno

```
mod5<-update(mod.ARMA30, weights = varIdent(form=~ 1 | N))
```

# Azufre

```
mod6<-update(mod.ARMA30, weights = varIdent(form=~ 1 | S))
```

**# Nitrógeno\*Azufre**

```
mod7<-update(mod.ARMA30, weights = varIdent(form=~ 1 | N*S))
```

**# Localidad:Nitrógeno**

```
mod8<-update(mod.ARMA30, weights = varComb(varIdent(form=~ 1 |
Localidad), varIdent(form=~ 1 |N)))
```

# Para la variable "dpa" dado que es continua podemos usar otras funciones

```
mod9<-update(mod.ARMA30, weights = varFixed(~dpa))
```

```
mod10<-update(mod.ARMA30, weights = varPower(form=~dpa))
```

```
mod11<-update(mod.ARMA30, weights = varExp(form=~dpa))
```

**# Comparamos los modelos**

```
library("bbmle")
```

```
AICtab(mod3, mod.ARMA30, mod4, mod5, mod6, mod7, mod8, mod9, mod10,
mod11, weights = T, delta = T, base = T, sort = T)
```

	AIC	dAIC	df	weight
mod8	5166.4	0.0	26	0.608
mod5	5168.6	2.2	23	0.201
mod4	5170.9	4.5	25	0.064
mod.ARMA30	5171.7	5.3	22	0.042
mod7	5172.1	5.8	25	0.034
mod6	5173.5	7.1	23	0.018
mod10	5173.5	7.2	23	0.017
mod11	5173.6	7.3	23	0.016
mod9	5179.3	12.9	22	<0.001
mod3	5184.9	18.5	19	<0.001

# El modelo con la función de la varianza que estima una varianza para los grupos de Localidad y Fertilización con nitrógeno: mod8,

# tiene la mejor bondad de ajuste.

# Observar que la diferencia en el AIC respecto del modelo sin función de la varianza (mod.ARMA30) y, con el modelo sin estructura de correlación (mod3).

**# 7.d.1- Verificamos la validez del modelo****# Autocorrelación**

```
plot(ACF(mod8, resType="normalized", na.action=na.omit), alpha=0.05,
grid=TRUE)
```

```
plot(ACF(mod.ARMA30, resType="normalized", na.action=na.omit), alpha=0.05,
grid=TRUE)
```

```
plot(ACF(mod3, resType="normalized", na.action=na.omit), alpha=0.05,
grid=TRUE)
```

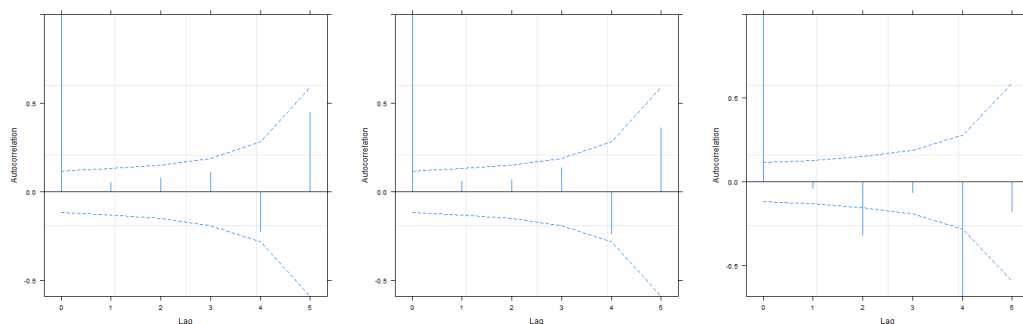


Figura 8. 11: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para los modelos "mod8" (Izq.), "mod.ARMA30" (Centro) y "mod3" (Der.). Las líneas llenas verticales indican



el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.

```
ACF(mod8,resType="normalized")
ACF(mod.ARMA30,resType="normalized")
ACF(mod3,resType="normalized")
```

#### # 7.d.2- Distribución de los residuales

# Residuales vs. las demás variables predictoras

```
layout(matrix(1:1,1,1))
Er<-resid(mod8, type="normalized")
Fit<-fitted(mod8)
plot(x=Fit, y=Er, xlab="ajustados", ylab="residuales estandarizados")
abline(a=0, b=0, col="red", lw=2)
```

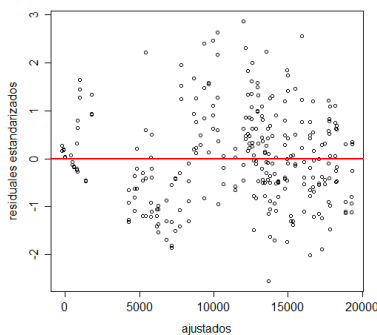


Figura 8. 12: residuales normalizados en relación a los valores ajustados por el modelo "mod8". La línea roja horizontal indica valores de cero para los residuales.

# El patrón en "U" y los predichos negativos casi hay desaparecido.

```
# Residuales en función de las variables predictoras
layout(matrix(1:4,2,2,byrow=T))
boxplot(Er~horde2$Localidad, main="Localidad", ylab="Residuales estandarizados", xlab="Localidad")
boxplot(Er~horde2$N, main="N", ylab="Residuales estandarizados", xlab="Fertilización")
boxplot(Er~horde2$S, main="S", ylab="Residuales estandarizados", xlab="Fertilización")
plot(Er~horde2$dpa, main="Dias post antesis", ylab="Residuales estandarizados", xlab="Días post antesis")
abline(a=0,b=0, col="red")
```

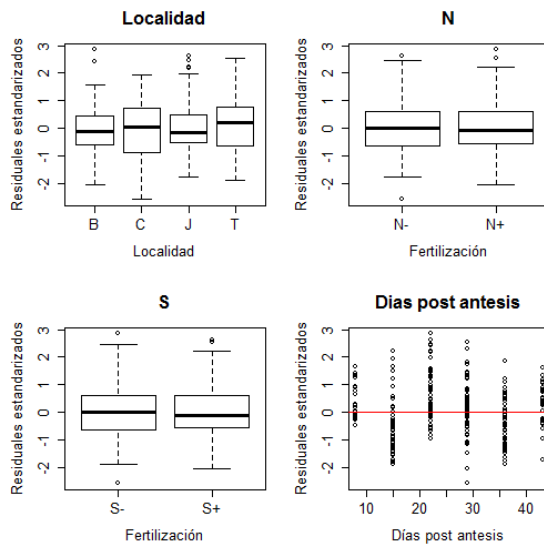


Figura 8. 13: residuales normalizados del modelo "mod8" en relación a los predictores: Localidad, Parcela, Fertilización con nitrógeno, Fertilización con azufre y días post antesis. De izquierda a derecha. La línea roja horizontal indica valores de cero para los residuales.

```
# Distribución normal de los residuos
qqnorm(Er, main="modelo")
qqline(Er)
```

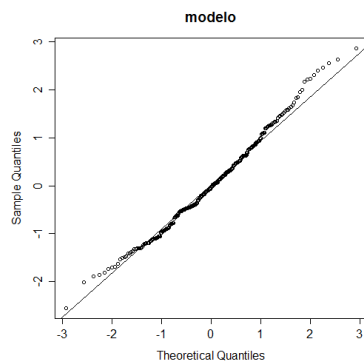


Figura 8. 14: gráfico de normalidad (cuantiles teóricos) para el modelo "mod8".

## # 8- Inferencia sobre los efectos fijos

### # 8.a- Generamos múltiples modelos

```
# Cambiamos al modo de máxima verosimilitud (ML)
```

```
modML<-update(mod9, method="ML")
```

```
# Pueden ocurrir dos tipos de no convergencia: fallas al estimar los parámetros
o, sobreestimación de los errores (ó, IC) de los parámetros (este último puede
ocurrir sin mensaje de advertencia)
```

```
# La solución para ambos tipos de error es reducir el tamaño y/o la complejidad
del modelo global (p.ej.: remover interacciones débiles).
```

```
# ?lmeControl
```

# Dentro de los argumentos posibles están maxIter, msMaxIter, niterEM, msMaxEval que sirven para aumentar por ejemplo, la cantidad de iteraciones por defecto.

```
ctrl <- lmeControl(maxIter= 5000, msMaxIter= 500)
```

```
# maxIter
```

# indica el máximo número de iteraciones para el algoritmo de iteraciones. Por defecto son 50

```
#
```

```
# msMaxIter
```

# indica el máximo número de iteraciones para los pasos de optimización dentro de la optimización de lme. Por defecto son 50.

```
mod8b<-update(mod8, control= ctrl)
```

```
modML<-update(mod8b, method="ML")
```

```
# Veamos como es el modelo
```

```
modML$call
```

```
lme.formula(fixed = Hord_B ~ N * S * dpa * I(dpa^2), data = horde2,
  random = ~1 | Localidad/Parcela, correlation = corARMA(p = 3, q = 0
  , form = ~1 | Localidad/Parcela), weights = varComb(varIdent(form =
  ~1 | Localidad), varIdent(form = ~1 | N)), method = "ML", control =
  ctrl)
```

```
library("MuMIn")
```

```
multi<-dredge(modML)
```

# Este proceso puede demorar unos minutos dado que debe estimar gran cantidad de modelos.

```
nrow(multi)
```

```
[1] 167
```

# Observar, la función "dredge" generó 167 modelos a partir de los predictores incluidos en el modelo global.

### # 8.b- Ordenar los modelos según su bondad de ajuste

# La función "model.sel" ordena los modelos según el ajuste (AIC, BIC, AICc, etc.).

# Además, genera una ponderación del ajuste relativa a los modelos incluido ("weight").

# Por defecto ordena los modelos según su AICc, un índice de AIC que corrige cuando el número de observaciones es bajo.

# El argumento "rank" permite cambiar el criterio de ajuste

```
RankAIC <- model.sel(multi, rank="AIC")
```

```
RankAIC
```

```
Global model call: lme.formula(fixed = Hord_B ~ N * S * dpa * I(dpa^
2), data = horde2, random = ~1 | Localidad/Parcela, correlation = co
rARMA(p = 3, q = 0, form = ~1 | Localidad/Parcela), weights = varCom
b(varIdent(form = ~1 | Localidad), varIdent(form = ~1 | N)), method
= "ML", control = ctrl)
```

```
---
```

```
Model selection table
```

	(Int)	dpa	dpa^2	N	S	dpa:I(dpa^2)	dpa:N	I(dpa^2):N	df	logLik
56	-7382	927.7	-2.538	+		-0.1774	+		16	-2608.
64	-7659	928.8	-2.588	+	+	-0.1767	+		17	-2607.
184	-6890	880.8	-1.583	+		-0.1782	+	+	17	-2607.
40	-9421	1239.0	-16.110	+			+		15	-2609.

```
48 -9693 1238.0 -16.100 + + + 16 -2609.
192 -7172 882.8 -1.648 + + -0.1775 + + 18 -2607.
```

```
      AIC      delta weight
612 5249.2 0.00 0.234
866 5249.7 0.51 0.181
939 5249.9 0.65 0.169
979 5250.0 0.73 0.162
218 5250.4 1.21 0.128
225 5250.5 1.23 0.127
```

```
Random terms (all models):
'1 | Localidad', '1 | Parcela %in% Localidad'
```

```
# Guardamos el resultado de los 256 modelos en una archivo
write.table(RankAIC, file = "13_Hordeinas_Ranking de Modelos.xls", append = F,
sep = "\t")
# IMPORTANTE: al abrir el archivo de excel, corregir la primer fila. Deben correr
un lugar a la derecha las columnas.
# La primera columna identifica el número del modelo y la segunda la el valor
estimado de las intercepciones.
# Las siguientes columnas muestran los valores estimados para los siguientes
predictores cuantitativos y un "+" cuando es un factor.
# Las últimas cinco columnas muestran:
#- la cantidad de parámetros del modelo (df)
#- el log de la máxima verosimilitud (logLik)
#- el AIC
#- el delta
#- el weight

# Observamos los primeros
head(RankAIC)
# Observar que ahora el weights está calculado entre los modelos que muestra la
función "head".
# Los mejores modelos incluyen a los predictores dpa, dpa^2, N y la interacción
dpa:N.
# Los tres mejores incluyen además, la interacción dpa:dpa^2

rel_impor<-importance(multi)
# La función "importance" calcula para cada variable predictora la suma de los
pesos relativos de AIC (weights) de los modelos que incluyen a la variable.
barplot(rel_impor[1:8], main= "Importancia relativa de las variables", las=2,
col=rainbow(15))

# La comparación es válida si las variables aparecen en la misma cantidad de
modelos.
# Entonces estaría bien comparar efectos principales entre sí (dpa, N y S),
interacciones dobles entre sí.
# Pero no, efectos principales vs. interacciones ya que las interacciones aparecen
en menos modelos.
# El gráfico muestra que las variables predictoras de mayor importancia son N
(fertilización con N), dpa y dpa^2 (días post antesis). Luego, S (Fertilización con
Azufre).
```

# Dentro de las interacciones, dpa:N y dpa^2:N son las más importante.  
 # Cuanto mayor es el valor, más importante es la variable en relación a las otras variables usadas en los mismos modelos.

### # 8.c- Identificar el mejor modelo

```
RankAIC[1,]
---
Model selection table
  (Int)   dpa dpa^2   N dpa:I(dpa^2) dpa:N df logLik   AIC  delta
56 -7382 927.7 -2.538 + -0.1774      +   16 -2608.6 5249.2    0
Random terms (all models):
'1 | Localidad', '1 | Parcela %in% Localidad'
```

# El modelo con mejor bondad de ajuste es el modelo 1208 que incluyo a los predictores:

```
# dpa
# dpa^2
# N
# dpa:I(dpa^2)
# dpa:N
```

```
AIC_BEst<-get.models(RankAIC,subset = delta <= 0)
```

```
AIC_BEst
$`56`
Linear mixed-effects model fit by maximum likelihood
Data: horde2
Log-likelihood: -2608.612
Fixed: Hord_B ~ dpa + I(dpa^2) + N + dpa:I(dpa^2) + dpa:N + 1
(Intercept)   dpa   I(dpa^2)   NN+   dpa:I(dpa^2)   dpa:NN+
-7382.20    927.6   -2.53    764.9    -0.177    81.56
```

```
Random effects:
Formula: ~1 | Localidad
(Intercept)
StdDev:    589.8462

Formula: ~1 | Parcela %in% Localidad
(Intercept) Residual
StdDev:    0.5471837 1706.745
```

```
Correlation Structure: ARMA(3,0)
Formula: ~1 | Localidad/Parcela
Parameter estimate(s):
      Phi1      Phi2      Phi3
0.33539080 -0.06172228 0.46488221
```

```
Combination of variance functions:
Structure: Different standard deviations per stratum
Formula: ~1 | Localidad
Parameter estimates:
      B      C      J      T
1.000000 1.307396 1.277289 1.445234
Structure: Different standard deviations per stratum
Formula: ~1 | N
Parameter estimates:
      N-      N+
1.000000 1.295295
Number of Observations: 287
Number of Groups:
      Localidad Parcela %in% Localidad
      4          60
```

**# 8.d- Identificar el modelo nulo**

```
RankAIC["1",]
Model selection table
  (Int) df    logLik      AIC    delta  weight
1 8371 11 -2795.841  5613.7  364.46      1
Random terms (all models):
'1 | Localidad', '1 | Parcela %in% Localidad'
```

# **ATENCIÓN:** el modelo nulo es el modelo llamado "1".

# El mejor modelo es el que se encuentra en primer lugar (la primera fila) dado que los ordenamos con este criterio

# mediante la función "model.sel".

**# 8.e- Verificar el mejor modelo**

# Para poder verificar el modelo, tenemos que calcular los residuales.

# Para ello, escribimos el modelo con la función lme

```
Best<-lme(Hord_B ~ dpa + dpa^2 + N + dpa:I(dpa^2) + dpa:N,
  random = ~1 | Localidad/Parcela,
  correlation = corARMA(p = 3, q = 0, form = ~1 | Localidad/Parcela),
  weights = varComb(varIdent(form = ~1 | Localidad),
    varIdent(form = ~1 | N)),
  method = "REML",
  control = ctrl,
  data = horde2)
```

**# 8.e.1- Verificamos la validez del modelo**

# Residuales vs. las demás variables predictoras

```
layout(matrix(1:6,2,3,byrow = TRUE))
```

```
ErB<-resid(Best, type="normalized")
```

```
FitB<-fitted(Best)
```

```
plot(x=FitB, y=ErB, xlab="ajustados", ylab="residuales estandarizados")
```

```
abline(a=0, b=0, col="red", lw=2)
```

**# 8.e.2- Residuales en función de las variables predictoras**

```
boxplot(ErB~horde2$Localidad, main="Localidad", ylab="Residuales
estandarizados", xlab="Localidad")
```

```
boxplot(ErB~horde2$N, main="N", ylab="Residuales estandarizados",
xlab="Fertilización")
```

```
boxplot(ErB~horde2$S, main="S", ylab="Residuales estandarizados",
xlab="Fertilización")
```

```
plot(ErB~horde2$dpa, main="Dias post antesis", ylab="Residuales
estandarizados", xlab="Días post antesis")
```

```
abline(a=0,b=0, col="red")
```

# Distribución normal de los residuos

```
qqnorm(ErB, main="modelo")
```

```
qqline(ErB)
```

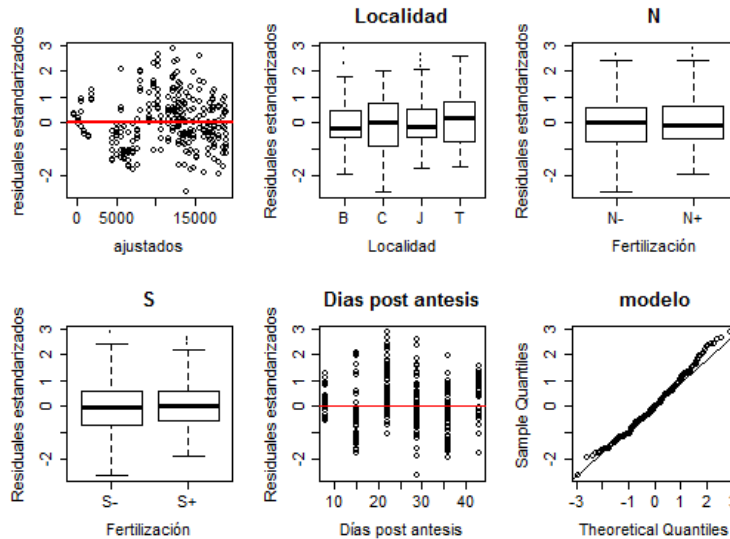


Figura 8. 15: residuales normalizados del modelo "Best" en relación a los ajustados por el modelo, los predictores: Localidad, Fertilización con nitrógeno, Fertilización con azufre, y días post antesis. Gráfico de normalidad (cuantiles teóricos) para el mismo modelo.

```
plot(ACF(Best,resType="normalized"),alpha=0.05,
grid=TRUE,form=~dpa|Localidad/Parcela)
```

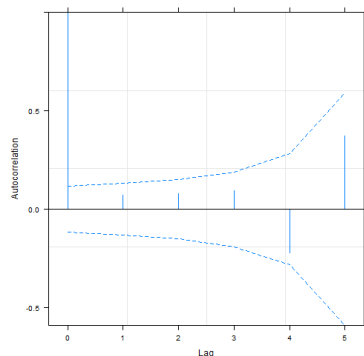


Figura 8. 16: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Best". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.

```
# Ajustados en función de los observados
layout(matrix(1:1,1,1))
plot(horde2$Hord_B, FitB, xlab="Observados", ylab="Ajustados")
abline(a=0, b=1, lwd=2, col="red") # agregamos una línea con ordenada al origen
= 0 y pendiente = 1
```

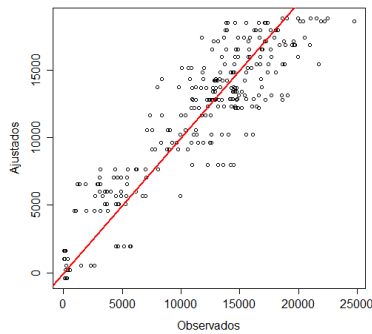


Figura 8. 17: residuales normalizados en relación a los valores ajustados por el modelo "Best". La línea roja horizontal indica valores de cero para los residuales.

### # 9- Gráfico del modelo

# Volvemos al modelo estimado por REML. Recordemos que "average" fue estimado por el método de REML

```

cl<-with(horde2, length(levels(N)))
#horde2$trat<-interaction(horde2$N,horde2$S)

layout(matrix(1:1,1,1))
plot(x=horde2$dpa, y = horde2$Hord_B,
     type = "p",lwd=2 ,col="gray",
     ylab="cantidad de horteinas B", xlab="días desde antesis (log)")
# type="n" indica que el gráfico esta vacío.
# Agregamos los puntos con un color separado por cada tratamiento de
fertiización
fixef(Best)

# N-
points(horde2$dpa[horde2$N == "N-"],horde2$Hord_B[horde2$N == "N-"],
       col = "black", pch = 16)

curve(fixef(Best)[1]+
      fixef(Best)[2]*(x)+
      fixef(Best)[4]*(x)*(x)^2,
      add=T, col="black", lw=2 )

# N+
points(horde2$dpa[horde2$N == "N+"],horde2$Hord_B[horde2$N == "N+"],
       col = "blue", pch = 16)

curve(fixef(Best)[1]+
      fixef(Best)[3]+
      fixef(Best)[2]*(x)+
      fixef(Best)[4]*(x)*(x)^2+
      fixef(Best)[5]*(x),
      add=T, col="blue", lw=2 )

```



```
legend("topleft",levels(horde2$N), col=c("black", "blue"), pch=16, bty="n")
```

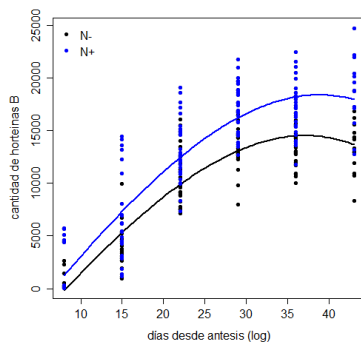


Figura 8. 18: cantidad de hordeínas B en función de los días post antesis [log(dpa)]. los puntos indican los valores observados y las líneas, el ajuste general del modelo para los tratamientos con (N+, negro) y sin (N- azul) fertilización con nitrógeno.

## # 10- Resumen

- # Escribimos el modelo de efectos mixtos que contempla la estructura dada por el diseño del experimento y el método de muestreo.
- # Observamos el patrón en los residuales del modelo.
- # Evaluamos la correlación dada por las medidas repetidas en el tiempo mediante la "función de auto-correlación" (ACF).
- # Encontramos la estructura óptima para los componentes aleatorios mediante funciones de correlación y, funciones de la varianza.
- # Evaluamos el componente fijo del modelo en este caso mediante inferencia multimodelo.
- # Finalmente, presentamos el modelo utilizando el método REML de estimación.

## # 11- Tarea

### # Análisis del modelo

- # a- Describir los componentes fijos y aleatorios del modelo.
- # b- Escribir el modelo con la nomenclatura de Gellman y Hill (2007).

```
Best$call
```

```
lme.formula(fixed = Hord_B ~ dpa + dpa^2 + N + dpa:I(dpa^2) + dpa:N,
  data = horde2, random = ~1 | Localidad/Parcela, correlation = corAR
  MA(p = 3, q = 0, form = ~1 | Localidad/Parcela), weights = varComb(v
  arIdent(form = ~1 | Localidad), varIdent(form = ~1 | N)), method = "
  REML", control = ctrl)
```

```
# c- Identificar:
```

- # - el parámetro que indica la bondad de ajuste.
- # - la variabilidad no explicada por el modelo entre Localidades
- # - la variabilidad no explicada por el modelo entre Parcela
- # - la variabilidad residual del modelo
- # - el parámetro que estima la autocorrelación
- # - los parámetros que estiman las varianzas para las localidades y niveles de fertilización

```
summary(Best)
```

```
Linear mixed-effects model fit by REML
Data: horde2
```

```

      AIC      BIC    logLik
5211.533 5266.162 -2590.767

Random effects:
Formula: ~1 | Localidad
(Intercept)
StdDev:    752.5143

Formula: ~1 | Parcela %in% Localidad
(Intercept) Residual
StdDev:    0.5908191 1732.243

Correlation Structure: ARMA(3,0)
Formula: ~1 | Localidad/Parcela
Parameter estimate(s):
  Phi1      Phi2      Phi3
0.3440516 -0.0636273 0.4552495
Combination of variance functions:
Structure: Different standard deviations per stratum
Formula: ~1 | Localidad
Parameter estimates:
      B      C      J      T
1.000000 1.299417 1.274500 1.431864
Structure: Different standard deviations per stratum
Formula: ~1 | N
Parameter estimates:
      N-      N+
1.000000 1.287687
Fixed effects: Hord_B ~ dpa + dpa^2 + N + dpa:I(dpa^2) + dpa:N
              Value Std.Error DF   t-value p-value
(Intercept) -7016.360  709.2470 224   -9.892689 0.0000
dpa          870.294   31.7040 224   27.450589 0.0000
NN+         766.596   734.8974  55    1.043133 0.3015
dpa:I(dpa^2) -0.211     0.0136 224  -15.521458 0.0000
dpa:NN+      81.619    22.5273 224    3.623116 0.0004
Correlation:
              (Intr) dpa      NN+      d:I(^2)
dpa          -0.714
NN+         -0.393  0.211
dpa:I(dpa^2) 0.549 -0.899  0.000
dpa:NN+      0.308 -0.269 -0.782  0.001

Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-2.65606114 -0.66486500 -0.07518114  0.61626164  2.64860662

Number of Observations: 287
Number of Groups:
      Localidad Parcela %in% Localidad
              4              60

```

# d- Interprete el siguiente resultado: ¿Qué indican/representan los valores estimados?

```

ranef(Best)$Localidad
(Intercept)
B    -248.9298
C     675.3857
J     345.9144
T    -772.3703

```

## # 12- Agradecimiento

# Datos gentilmente provistos por Andrés Peton y Eduardo Pagano de la Facultad de Agronomía de la Universidad de Buenos Aires. Peton, A. "Patrón de acumulación de hordeínas en granos de cebada cervecera y su relación con la fertilización azufrada y nitrogenada". Tesis de maestría en curso, EPG-FAUBA.

Por razones didácticas, los datos presentados son un subconjunto modificado de los datos originales.

Datos jerárquicos

# Correlación espacial

## Contenidos

<b># CAPÍTULO 9: Correlación espacial .....</b>	<b>149</b>
<b># 1- Configuración inicial.....</b>	<b>149</b>
<b># 2- El caso: Interacciones entre arbustos mediadas por el fuego.....</b>	<b>149</b>
# 2.a- Pregunta de interés .....	149
# 2.b- Variable respuesta.....	149
# 2.c- Variable/s predictoras de interés.....	150
<b># 3- Explorar los datos .....</b>	<b>150</b>
# 3.a- Datos ubicación.....	150
# 3.b- Datos densidad.....	151
# 3.c- Distribución espacial de las especies II .....	151
<b># 4- Construcción del modelo inicial .....</b>	<b>152</b>
# 4.a- Distribución espacial de los residuos.....	152
# 4.b- Variogramas .....	153
<b># 5- Modelamos la correlación espacial .....</b>	<b>154</b>
# 5.a- Generar modelos .....	154
# 5.b- Comparar los modelos .....	154
<b># 6- Verificación del mejor modelo .....</b>	<b>155</b>
# 6.a- Variograma.....	155
# 6.b- Distribución de los residuales.....	155
<b># 7- Modelar las varianzas.....</b>	<b>156</b>
# 7.a- Planteamos algunos modelos .....	156
# 7.b- Comparamos los modelos.....	156
<b># 8- Evaluación de los supuestos del modelo .....</b>	<b>157</b>
# 8.a- Variograma.....	157
# 8.b- Residuales .....	157
# 8.c- Distribución espacial de los residuos.....	157
# 8.d- Normalidad.....	157
<b># 9- Resumen.....</b>	<b>158</b>
<b># 10- Agradecimiento.....</b>	<b>158</b>

## # CAPÍTULO 9: Correlación espacial

### # 1- Configuración inicial

# Definir el directorio de trabajo  
# setwd("D:\\Mis documentos")

### # 2- El caso: Interacciones entre arbustos mediadas por el fuego

El marco de este experimento es estudiar las condiciones que favorecen la regeneración de arbustos luego de un incendio. El sitio de estudio se encuentra en el Parque Nacional al Palmar en la provincia de Entre Ríos (Fig. 9.1).

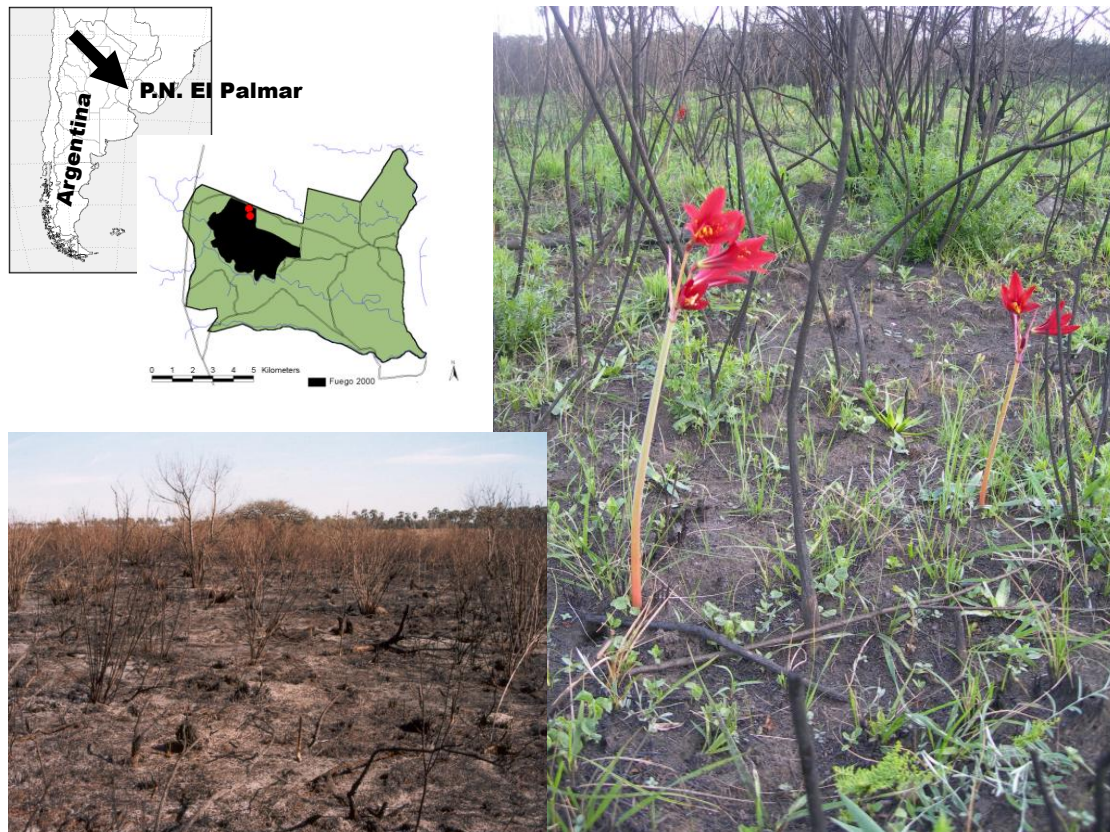


Figura 9. 1: ubicación del sitio de estudio y fotos ilustrando la situación post-incendio y, durante la regeneración.

Se mapearon todos los arbustos de *Baccharis dracunculifolia* y de *Eupatorium buniifolium* inmediatamente luego de un incendio. Para cada individuo se determinó la especie, el tamaño y el estado (plántula, adulto, muerto). Además, se mapearon todas las plántulas que se instalaron en el año posterior al incendio. Se caracterizó la microtopografía del sitio de estudio (con una precisión de 40 cm aprox.).

#### # 2.a- Pregunta de interés

# ¿Ocupan las plántulas de *Baccharis dracunculifolia* el espacio liberado por individuos muertos luego de un incendio?

#### # 2.b- Variable respuesta

# Densidad de plántulas

### # 2.c- Variable/s predictoras de interés

# Densidad de individuos adultos muertos  
# Topografía

### # 3- Explorar los datos

# NO diagnóstico, solo a modo exploratorio

#### # 3.a- Datos ubicación

# Leemos una tabla de datos con la ubicación en x e y de individuos de las dos especies de plantas clasificadas en adultos y plántulas

```
ubi<-read.csv("9_Ubicacion.csv", header=T)
summary(ubi)
```

```
with(subset(ubi,sp=="Ebplantulas"),plot(x,y))
title(main=expression(paste(italic("Eupatorium buniifolium")," plántulas")))
```

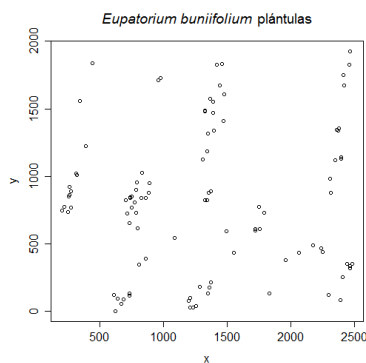


Figura 9. 2: distribución espacial de plántulas *Eupatorium buniifolium*

```
with(subset(ubi,sp=="Ebrebrotadas"),plot(x,y))
title(main=expression(paste(italic("Eupatorium buniifolium")," rebrotados")))
with(subset(ubi,sp=="Bdplantula"),plot(x,y))
title(main=expression(paste(italic("Baccharis dracunculifolia")," plántulas")))
```

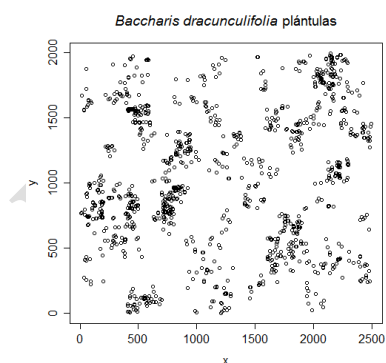


Figura 9. 3: distribución espacial de plántulas *Baccharis dracunculifolia*

```
with(subset(ubi,sp=="Bdadulto"),plot(x,y))
title(main=expression(paste(italic("Baccharis dracunculifolia")," adultos")))
```

```
with(subset(ubi,sp=="Ebmueras"),plot(x,y))
title(main=expression(paste(italic("Eupatorium buniifolium")," adultos muertos")))
```



# Observar la distribución espacial de las distintas especies  
# de adultos, plántulas o muertos

### # 3.b- Datos densidad

```
# Leemos una segunda tabla con los datos de densidad en la posición
# topográfica relativa de la parcela dividida en una grilla de 20 x 25 metros
densi<-read.table("9_Densidad.csv", header=T, sep=",")
summary(densi)
# Esta tabla contiene las especies, la posición (xx e yy) y la posición topográfica.
library("lattice")
wireframe( densi$microtop ~ densi$xx * densi$yy,main="Topografía",
xlab="Posición x", ylab="Posición y", zlab="Altura")
```

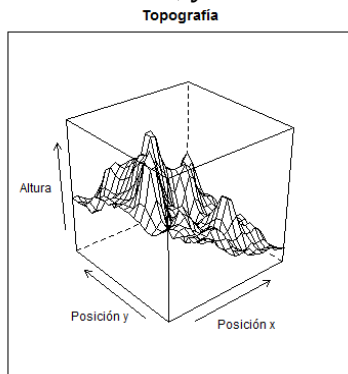


Figura 9. 4: topografía de sitio de estudio

```
wireframe( densi$Bdplantula ~ densi$xx * densi$yy,main="Baccharis plántula",
xlab="Posición x", ylab="Posición y", zlab="Densidad")
```

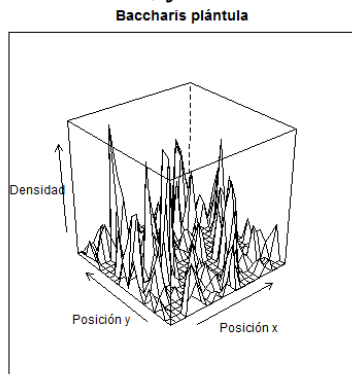


Figura 9. 5: distribución espacial de la densidad de plántulas *Baccharis dracunculifolia*

```
wireframe( densi$Bdadulto ~ densi$xx * densi$yy,main="Baccharis adulto",
xlab="Posición x", ylab="Posición y", zlab="Densidad")
```

# **TAREA** : hacer gráficos para densidad y distribución de *Eupatorium*.

### # 3.c- Distribución espacial de las especies II

# Estas figuras las podemos girar seleccionandolas y moviendo con el mouse.

```
library("rgl")
plot3d(densi$xx,densi$yy, densi$microtop, type="s",col="red", size=2,
main="Topografía", xlab="posicion x", ylab="posicion y", zlab="Altura")
```

```
plot3d(densi$xx,densi$yy, densi$Bdplantula, type="h",col="green",
size=3,main="Baccharis plantula", xlab="posicion x", ylab="posicion y",
zlab="Densidad")
```

```
library("sp")
```

```
# Creamos un objeto ("plántulas") para poder graficar con bubble
plantulas<- data.frame(densi$Bdplantula, densi$xx, densi$yy)
coordinates(plantulas) <- c("densi.xx","densi.yy")
```

```
bubble(plantulas, "densi.Bdplantula", col = c("black","red"),
main = "Densidad plantulas de Baccharis", xlab = "Coordenadas x", ylab =
"Coordenadas y",
key.entries = 2^(0:5))
```

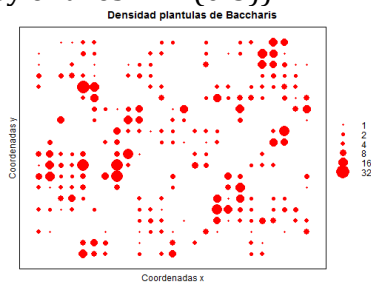


Figura 9. 6: distribución espacial de la densidad de plántulas *Baccharis dracunculifolia*

#### # 4- Construcción del modelo inicial

```
# En este caso, los datos no poseen una estructura jerárquica.
```

```
# Por lo tanto, en este caso, no hay factores aleatorios
```

```
# y no es adecuado (posible) utilizar la función lme.
```

```
# Utilizaremos la función gls (Generalized Least Squares)
```

```
library(nlme)
```

```
den.gls<- gls(Bdplantula ~ Bdadulto+Ebmueratas+microtop, data=densi)
```

```
summary(den.gls)
```

#### # 4.a- Distribución espacial de los residuos

```
E1<-residuals(den.gls, type="normalized")
```

```
F1<-fitted(den.gls)
```

```
mydata<- data.frame(E1, densi$xx, densi$yy)
```

```
colnames(mydata) <- c("E1", "xx", "yy")
```

```
coordinates(mydata) <- ~xx+yy
```

```
bubble(mydata, "E1", col = c("black","red"),
main = "Residuales", xlab = "Coordenadas x",
ylab = "Coordenadas y")
```

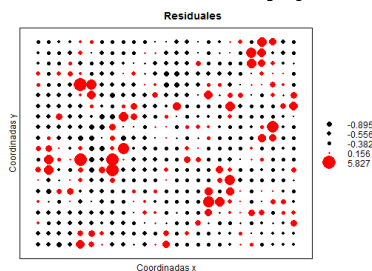




Figura 9. 7: distribución espacial de los residuales normalizados del modelo "den.gls". Los puntos rojos indican valores positivos y los negro indican valores negativos. El tamaño de los puntos indica la magnitud del residual.

# Los puntos negros en el gráfico indican valores negativos y los puntos rojos, positivos.

# Una distribución deseable de los residuos no debería mostrar patrones de distribución espacial.

# (Ej.: valores positivos más importantes ó, predominancia de algún sector)

#### # 4.b- Variogramas

```
library(gstat)
coord<- data.frame(densi$xx, densi$yy)
colnames(coord) <- c("xx", "yy")
coordinates(coord) <- ~xx+yy
gridded(coord) <- TRUE
```

```
Vario <- variogram(E1 ~ 1, coord, width=.5)
plot(Vario)
```

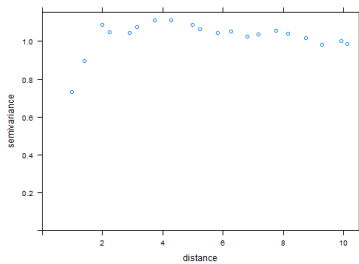


Figura 9. 8: semivariograma de los residuales normalizados del modelo "den.gls"

```
Vario1 <- variogram(E1 ~ 1, coord, width=.5, alpha = c(0, 45, 90, 135))
plot(Vario1)
```

# El argumento "alpha" modifica la dirección en el plano (x,y) en grados positivos en sentido horario (0 = y positivo).

# En ausencia de correlación espacial, todos los valores del variograma deberían estar cercanos a 1 (ojo, con residuos normalizados).

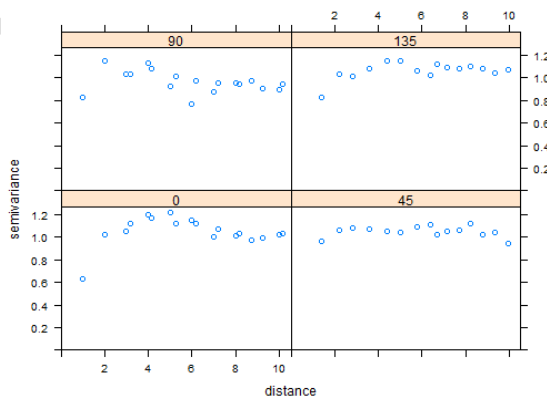


Figura 9. 9: semivariograma de los residuales normalizados del modelo "den.gls" en cuatro direcciones en el plano (0, 45, 90 y 135 °C).

# En este caso observamos que los valores caen a medida que la distancia se reduce.

# Es decir, observamos correlación espacial entre los datos.

# Cuanto más cerca están en el espacio son más parecidos

# (menor semi-varianza).

## # 5- Modelamos la correlación espacial

### # 5.a- Generar modelos

# Incluir en el modelo inicial una estructura de correlación.

```
# corLin correlación espacial lineal
# corExp correlación espacial exponencial
# corSpher correlación espacial esférica
# corGaus correlación espacial gaussiana
# corRatio correlación espacial cuadrática
? corClasses
```

```
# ¡ATENCIÓN!: Estos modelos pueden demorar en correr
den.gls2<- update(den.gls,correlation=corLin(form = ~ xx + yy))
E2<-residuals(den.gls2, type="n")
Vario2 <- variogram(E2 ~ 1, coord, width=.5, alpha = c(0, 45, 90, 135))
plot(Vario2)
```

```
den.gls3<- update(den.gls,correlation=corExp(form = ~ xx + yy))
E3<-residuals(den.gls3, type="n")
Vario3 <- variogram(E3 ~ 1, coord, width=.5, alpha = c(0, 45, 90, 135))
plot(Vario3)
```

```
den.gls4<- update(den.gls,correlation=corSpher(form = ~ xx + yy))
E4<-residuals(den.gls4, type="n")
Vario4 <- variogram(E4 ~ 1, coord, width=.5, alpha = c(0, 45, 90, 135))
plot(Vario4)
```

```
den.gls5<- update(den.gls,correlation=corGaus(form = ~ xx + yy))
E5<-residuals(den.gls5, type="n")
Vario5 <- variogram(E5 ~ 1, coord, width=.5, alpha = c(0, 45, 90, 135))
plot(Vario5)
```

```
den.gls6<- update(den.gls,correlation=corRatio(form = ~ xx + yy))
E6<-residuals(den.gls6, type="n")
Vario6 <- variogram(E6 ~ 1, coord, width=.5, alpha = c(0, 45, 90, 135))
plot(Vario6)
```

### # 5.b- Comparar los modelos

```
library("bbmle")
AICtab(den.gls,den.gls2,den.gls3,den.gls4,den.gls5,den.gls6,
       weights=T, sort=T, base=T)
```

AIC      dAIC      df weight

```
den.gls4 2747.0 0.0 6 0.9375
den.gls5 2752.5 5.5 6 0.0602
den.gls6 2759.6 12.6 6 0.0017
den.gls3 2762.2 15.2 6 <0.001
den.gls2 2766.6 19.6 6 <0.001
den.gls 2836.7 89.6 5 <0.001
```

# El modelo den.gls4 es el que presenta mejor ajuste dentro de este conjunto.  
# Observar el AIC, el delta respecto de los siguientes y el peso.

## # 6- Verificación del mejor modelo

### # 6.a- Variograma

```
E4<-residuals(den.gls4, type="n")
Vario4 <- variogram(E4 ~ 1, coord, width=.5)
plot(Vario4)
```

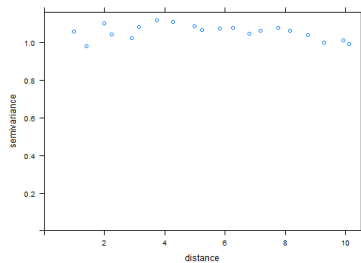


Figura 9. 10: semivariograma de los residuos normalizados del modelo "den.gls4"

# No observamos correlación espacial.  
# Los valores del variograma son similares para distintas distancias.

### # 6.b- Distribución de los residuales

```
F4<-fitted(den.gls4)
layout(matrix(1:4,2,2))
plot(F4, E4)
abline(0,0, lwd=3, col="red")
```

```
plot(densi$Bdadulto,E4)
abline(0,0, lwd=3, col="red")
plot(densi$Ebmueras,E4)
abline(0,0, lwd=3, col="red")
plot(densi$microtop,E4)
abline(0,0, lwd=3, col="red")
```

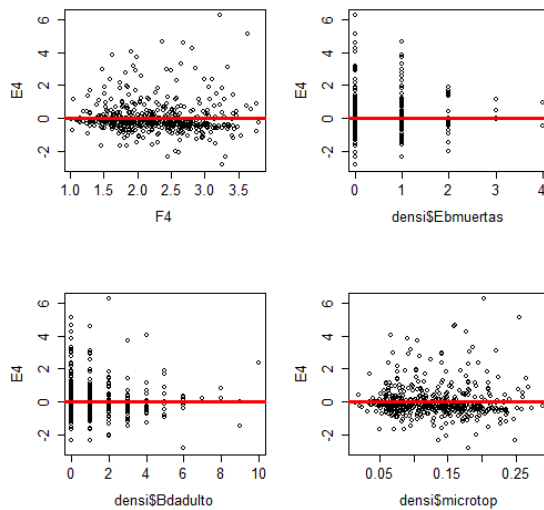


Figura 9. 11: residuales normalizados (E4) del modelo "den.gls4" en relación a los ajustados (F4) por el modelo y los predictores: densidad de *Eupatorium buniifolium* muertas (densi\$Ebmueratas), de *Baccharis dracunculifolia* adultos (densi\$adulto) y la topografía (densi\$microtop).

# Observar patrones en la distribución de los errores asociados a los valores predichos y, a las variables predictoras del modelo.

### # 7- Modelar las varianzas

```
densi$fEBM<-as.factor(densi$Ebmueratas)
```

#### # 7.a- Planteamos algunos modelos

```
den.gls4.1 <- update(den.gls4, weights = varIdent(~ 1|fEBM))
den.gls4.2 <- update(den.gls4, weights = varExp(form= ~ Bdadulto))
den.gls4.3 <- update(den.gls4, weights = varExp(form= ~ Ebmueratas))
den.gls4.4 <- update(den.gls4, weights = varExp(form= ~ microtop))
den.gls4.5 <- update(den.gls4, weights = varPower(form= ~ microtop))
den.gls4.6 <- update(den.gls4, weights = varFixed( ~ microtop))
```

# TAREA: pueden plantear otros modelos...

#### # 7.b- Comparamos los modelos

```
library("bbmle")
AICctab(den.gls4, den.gls4.1, den.gls4.2, den.gls4.3, den.gls4.4,
        den.gls4.5, den.gls4.6, weights=T, base=T, sort=T)
```

	AIC	dAIC	df	weight
den.gls4.6	2706.6	0.0	6	0.602
den.gls4.5	2707.7	1.1	7	0.350
den.gls4.4	2711.7	5.0	7	0.048
den.gls4.3	2745.1	38.5	7	<0.001
den.gls4	2747.0	40.4	6	<0.001
den.gls4.1	2747.0	40.4	6	<0.001
den.gls4.2	2747.3	40.7	7	<0.001

# El modelo den.gls4.6 con la función varFixed es el que presenta mejor ajuste dentro de este conjunto.

# Observar los valores de AIC, los delta y el peso del primer modelo.

## # 8- Evaluación de los supuestos del modelo

### # 8.a- Variograma

```
E4.6<-residuals(den.gls4.6, type="n")
Vario4.6 <- variogram(E4.6 ~ 1, coord)
plot(Vario4.6)
# No observamos correlación espacial.
# Los valores del variograma son similares a distancias contrastantes.
```

### # 8.b- Residuales

```
F4.6<-fitted(den.gls4.6)
layout(matrix(1:4,2,2))
plot(F4.6, E4.6)
abline(0,0, lwd=3, col="red")
plot(densi$Bdadulto,E4.6)
abline(0,0, lwd=3, col="red")
plot(densi$Ebmueratas,E4.6)
abline(0,0, lwd=3, col="red")
plot(densi$microtop,E4.6)
abline(0,0, lwd=3, col="red")
```

### # 8.c- Distribución espacial de los residuos

```
mydata4.6<- data.frame(E4.6, densi$xx, densi$yy)
coordinates(mydata4.6) <- c("densi.xx","densi.yy")
bubble(mydata4.6, "E4.6", col = c("black","red"),
       main = "Residuales", xlab = "Coordinadas x",ylab = "Coordinadas y")
# Los puntos negros en el gráfico indican valores negativos y los puntos rojos,
positivos.
# Una distribución deseable de los residuos no debería mostrarnos patrones de
distribución
# (Ej.: valores positivos más importantes ó, predominancia de algún sector)

# Si lo comparamos con el gráfico de distribución de residuales del modelo
inicial
# observamos que la distribución de positivos y negativos es más homogénea en
el último modelo.

# Modelo inicial:
bubble(mydata, "E1", col = c("black","red"),
       main = "Residuales", xlab = "Coordinadas x",
       ylab = "Coordinadas y")
```

### # 8.d- Normalidad

```
layout(matrix(1:1,1,1))
qqnorm(E4.6)
qqline(E4.6)
```

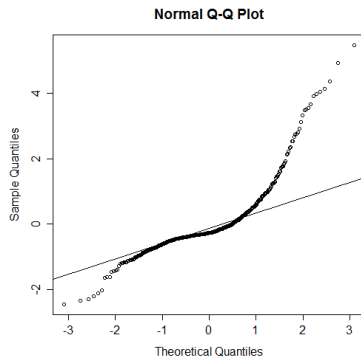


Figura 9. 12: Gráfico de normalidad (cuantiles teóricos) para el mismo modelo "den.gls4.6"

# Observamos que no cumplimos con el supuesto. El modelo no sería válido.

### # 9- Resumen

# En este caso:

# - practicamos numerosas formas de graficar datos espacialmente explícitos.

# - incorporamos al modelo estructuras de correlación espacial.

# En este caso, los supuestos del modelo no se cumplen,

# posiblemente esto se deba a que la variable "densidad"

# no se describa correctamente con una distribución normal...

# Retomaremos este ejemplo más adelante con otras distribuciones.

### # 10- Agradecimiento

# Datos provistos por Fernando Biganzoli y William B. Batista de la Facultad de Agronomía de la Universidad de Buenos Aires. Biganzoli, F., T. Wiegand & W. B. Batista. 2009. Fire-mediated interactions between shrubs in a South American temperate savannah. *Oikos* 118: 1383-1395. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.

# Correlación espacial - Intervalos desiguales

## Contenidos

<b># CAPÍTULO 10: Correlación espacial - Intervalos desiguales.....</b>	<b>160</b>
<b># 1- Configuración inicial.....</b>	<b>160</b>
# 1.a- Definir el directorio de trabajo .....	160
# 1.b- Leer el archivo de datos .....	160
<b># 2- El caso: Contenido de nitrógeno en suelos de Misiones.....</b>	<b>160</b>
# 2.a- Pregunta de interés .....	160
# 2.b- Variable respuesta.....	160
# 2.c- Variable/s predictoras de interés.....	161
# 2.d- Estructura de jerarquías en el diseño.....	161
# 2.e- Ordenar los datos .....	161
<b># 3- Explorar los datos .....</b>	<b>161</b>
<b># 4- Modelo inicial .....</b>	<b>161</b>
# 4.a- Efectos fijos del modelo .....	161
# 4.b- Modelo con estructura del diseño.....	161
# 4.c- Verificación del modelo .....	161
<b># 5- Correlación espacial.....</b>	<b>162</b>
# 5.a- Modelo con correlación .....	162
# 5.b- Comparación entre modelos .....	163
# 5.c- Verificación del modelo .....	163
<b># 6- Funciones de la varianzas.....</b>	<b>164</b>
# 6.a- Generar modelos con funciones de la varianzas .....	164
# 6.b- Comparación de modelos .....	164
# 6.c- Comparamos los variogramas.....	165
# 6.d- Verificación del mejor modelo.....	165
<b># 7- Ajustamos el componente fijo del modelo.....</b>	<b>166</b>
<b># 8- Modelos alternativos .....</b>	<b>167</b>
# 8.a- Generar modelos alternativos.....	167
# 8.b- Inferencia.....	167
# 8.c- Test de comparaciones múltiples .....	167
<b># 9- Gráficos del modelo final .....</b>	<b>168</b>
<b># 10- Resumen .....</b>	<b>169</b>
<b># 11- Series de tiempo .....</b>	<b>169</b>
<b># 12- Agradecimiento.....</b>	<b>169</b>

## # CAPÍTULO 10: Correlación espacial - Intervalos desiguales

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo de datos

```
suelo <- read.csv("10_Suelo.csv",header = TRUE)
```

#### # Alternativa:

```
# Abrir la planilla de datos desde la hoja de cálculos (LibreOffice, Excel, etc.).
```

```
# Copiar las celdas que contienen los datos y correr la sentencia:
```

```
# suelo=read.delim("clipboard",dec=".", header=TRUE)
```

```
# summary(suelo)
```

### # 2- El caso: Contenido de nitrógeno en suelos de Misiones

# Cambios en la calidad de la materia orgánica asociados al cambio en el uso de la tierra (Fig. 10.1). Contenido de nitrógeno en suelos de Misiones

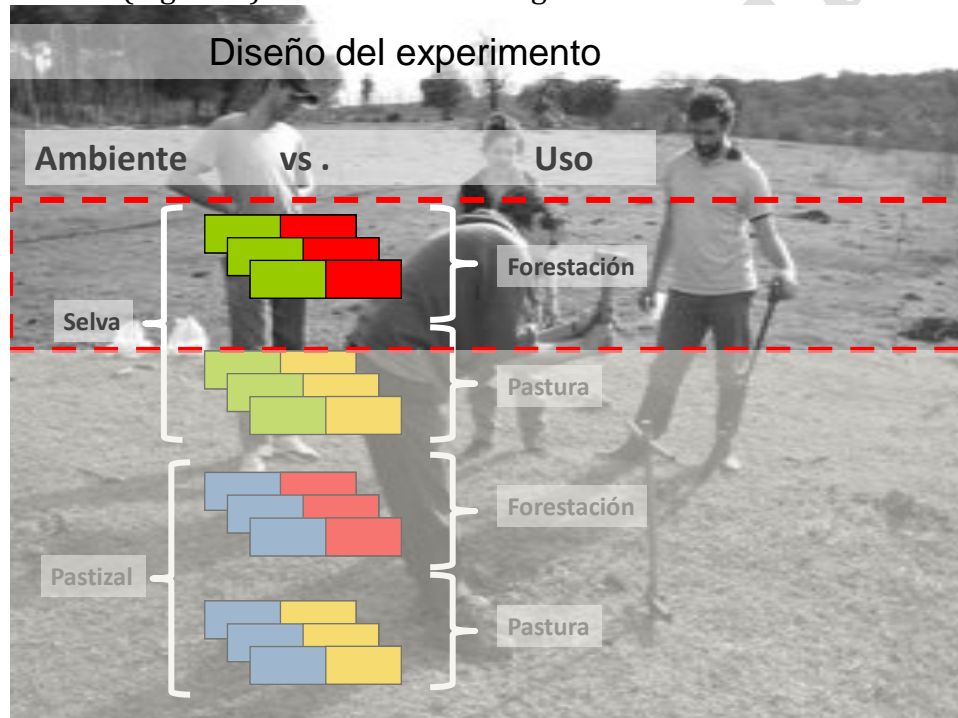


Figura 10. 1: esquema del diseño del experimento. El recuadro punteado rojo muestra la parte de los datos que utilizamos en este práctico.

#### # 2.a- Pregunta de interés

# ¿Qué efecto tiene el uso del suelo sobre la calidad (contenido de nitrógeno) de la materia orgánica del suelo?

#### # 2.b- Variable respuesta

# N mineral [g N.kg<sup>-1</sup> suelo] asociado a las fracciones más estables de la materia orgánica del suelo.

# En cada parcela se sacó un cilindro de suelo con un barreno de donde se cortaron siete extracciones para las determinaciones.



**# 2.c- Variable/s predictor de interés**

# - Uso: forestal o natural

**# 2.d- Estructura de jerarquías en el diseño**

# Determinaciones (7) profundidades dentro de cada parcela

# Parcelas (2) dentro de cada bloque

# Bloques 3

**# 2.e- Ordenar los datos**

```
suelo<-suelo[with(suelo,order(Bloque,Parcela, Prof)), ]
```

```
View(suelo)
```

```
suelo$Bloque<-as.factor(suelo$Bloque)
```

```
summary(suelo)
```

Bloque	Uso	Parcela	Prof	N
1:14	Forestal :21	selvaF1 :7	Min. : 5.00	Min. :0.9565
2:14	Natural :21	selvaF2 :7	1st Qu.: 10.00	1st Qu.:1.3666
3:14		selvaF3 :7	Median : 30.00	Median :1.6461
		selvaSF1:7	Mean : 40.71	Mean :1.7491
		selvaSF2:7	3rd Qu.: 70.00	3rd Qu.:2.0331
		selvaSF3:7	Max. :100.00	Max. :3.0443

**# 3- Explorar los datos**

```
plot(suelo$Uso, suelo$N,xlab = "Uso",ylab = "Nitrógeno (g.kg-1)")
```

**# 4- Modelo inicial**

```
library(nlme)
```

**# 4.a- Efectos fijos del modelo**

```
func<-formula(N ~ Uso)
```

**# 4.b- Modelo con estructura del diseño**

```
mrM0 <- lme(func, random=~1|Bloque/Parcela, data = suelo)
```

**# 4.c- Verificación del modelo****# 4.c.1- Residuales**

```
plot(mrM0)
```

```
Er<-residuals(mrM0, type="normalized")
```

```
Fit<-predict(mrM0)
```

```
plot(Fit, Er, xlab = "Predichos", ylab = "Residuos estandarizados")
```

```
abline(0,0, col="red", lwd=3)
```

```
plot(suelo$Uso, Er, xlab = "Uso", ylab = "Residuos estandarizados")
```

```
plot(suelo$Parcela, Er, xlab = "Parcelas", ylab = "Residuos estandarizados")
```

```
plot(suelo$Bloque, Er, xlab = "Bloques", ylab = "Residuos estandarizados")
```

**# 4.c.2- Normalidad**

```
qqnorm(Er, main="modelo")
```

```
qqline(Er)
```

**# 4.c.3- Observados vs. predichos**

```
plot(Fit, suelo$N, xlab = "Predichos", ylab = "Observados")
```

```
abline(0,1)
```

# Si bien consideramos la relación espacial al incorporar los efectos de los bloques y las parcelas, podría haber un patrón asociado a otra variable no incluido en el modelo.

```
# A continuación, observamos si la profundidad
```

```
# (o distancia) del cilindro genera algún patrón.
```

```
plot(suelo$Prof,Er, xlab = "Profundidad", ylab = "Residuos estandarizados")
```

```
abline(0,0, col="red", lwd=3)
```

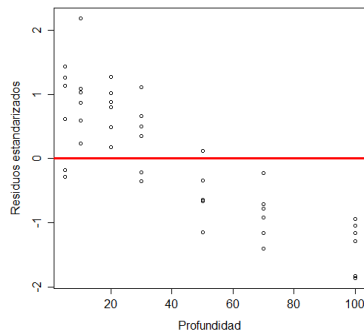


Figura 10. 2: residuales normalizados del modelo "mrM0" en relación a la profundidad.

# Este patrón en los residuos del modelo puede deberse a una correlación entre las extracciones del mismo barreno.

### # 5- Correlación espacial

# **IMPORTANTE:** cuando las medidas NO son equidistantes evaluamos

# la correlación a través de un (semi)variograma

```
library("nlme")
```

```
plot(Variogram(mrM0, form= ~ Prof|Bloque/Parcela,
```

```
resType="normalized", robust = T))
```

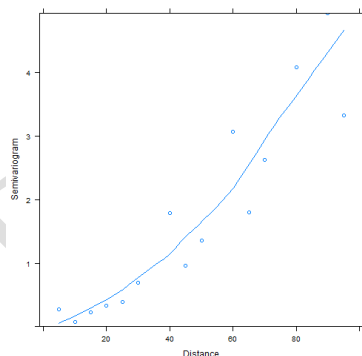


Figura 10. 3: semivariograma de los residuales normalizados del modelo "mrM0"

# En el variograma los valores entorno a 1 sin un patrón (más o menos horizontal) indican que no hay autocorrelación.

# En este caso, observamos un patrón que indica correlación espacial.

### # 5.a- Modelo con correlación

# Planteamos algunos modelos para corregir correlación

```
mrM1 <- update(mrM0, correlation = corExp(form = ~ Prof|Bloque/Parcela))
```

```
plot(Variogram(mrM1, form= ~ Prof|Bloque/Parcela, resType="normalized",
robust = T))
```

```
# Observamos que el patrón de correlación desapareció.
```

```
# Los valores están alrededor de 1
```

```
mrM2 <- update(mrM0, correlation = corSpher(form = ~ Prof|Bloque/Parcela))
plot(Variogram(mrM2, form= ~ Prof|Bloque/Parcela, resType="normalized",
robust = T))
```

```
mrM3 <- update(mrM0, correlation = corGaus(form = ~ Prof|Bloque/Parcela))
plot(Variogram(mrM3, form= ~ Prof|Bloque/Parcela, resType="normalized",
robust = T))
```

```
mrM4 <- update(mrM0, correlation = corRatio(form = ~ Prof|Bloque/Parcela))
plot(Variogram(mrM4, form= ~ Prof|Bloque/Parcela, resType="normalized",
robust = T))
```

```
mrM5 <- update(mrM0, correlation = corCAR1(form = ~ Prof|Bloque/Parcela))
plot(Variogram(mrM5, form= ~ Prof|Bloque/Parcela, resType="normalized",
robust = T))
```

#### # 5.b- Comparación entre modelos

```
library("bbmle")
```

```
AICtab(mrM0, mrM1, mrM2, mrM3, mrM4, mrM5,
```

```
weights = T, delta = TRUE, base=T, sort = TRUE)
```

	AIC	dAIC	df	weight
mrM2	37.3	0.0	6	0.68
mrM5	40.2	2.9	6	0.16
mrM1	40.2	2.9	6	0.16
mrM4	50.6	13.3	6	<0.001
mrM3	57.2	19.9	6	<0.001
mrM0	72.8	35.4	5	<0.001

```
# El modelo con estructura de correlación esférica ("mrM2") fue el que tuvo
mejor ajuste dentro del conjunto de modelos que comparamos.
```

#### # 5.c- Verificación del modelo

##### # 5.c.1- Residuales

```
Er2<-residuals(mrM2, type="normalized")
```

```
# Atención: en nlme el argumento es "type" y no "resType"
```

```
Fit2<-fitted(mrM2)
```

```
plot(Fit2, Er2, xlab = "Predichos",ylab = "Residuos estandarizados")
```

```
abline(0,0, col="red")
```

```
plot(suelo$Prof, Er2, xlab = "Profundidad", ylab = "Residuos estandarizados")
```

```
abline(0,0, col="red", lwd=3)
```

```
plot(suelo$Uso,Er2, xlab = "Uso", ylab = "Residuos estandarizados")
```

```
plot(suelo$Parcela,Er2, xlab = "Parcela", ylab = "Residuos estandarizados")
```

```
plot(suelo$Bloque,Er2, xlab = "Bloque", ylab = "Residuos estandarizados")
```

**# 5.c.2- Normalidad**

```
qqnorm(Er2, main="modelo")
qqline(Er2)
```

**# 5.c.3- Observados vs. predichos**

```
plot(Fit2, suelo$N, xlab = "Predichos", ylab = "Observados")
abline(0,1)
```

# A pesar que el modelo mrM2 mejoró el ajuste respecto del modelo sin correlación, aún observamos patrones en los residuales.

**# 6- Funciones de la varianzas****# 6.a- Generar modelos con funciones de la varianzas**

# Probamos algunas funciones de las varianzas que podrían corregir el/los patrones observados

```
mrM2.vI <- update(mrM2, weights=varIdent(form=~1|Bloque))
mrM2.vIB <- update(mrM2, weights=varIdent(form=~1|Uso))
# El modelo anterior no converge!
```

```
# Error in lme.formula(fixed = func, data = suelo, random = ~1 |Bloque/Parcela,
#           nlminb problem, convergence error code = 1
#           message = iteration limit reached without convergence (10)
#
```

# Pueden ocurrir dos tipos de no convergencia: fallas al estimar los parámetros o, sobreestimación de los errores (ó, IC) de los parámetros (este ultimo puede ocurrir sin mensaje de advertencia).

# La solución para ambos tipos de error es reducir el tamaño y/o la complejidad del modelo global (p.ej.: remover interacciones débiles).

```
# ?lmeControl
```

# Dentro de los argumentos posibles están maxIter, msMaxIter, niterEM, msMaxEval que sirven para aumentar por ejemplo, la cantidad de iteraciones por defecto.

```
ctrl <- lmeControl(maxIter= 5000)
ctrl2 <- lmeControl(opt="optim") # Le pedimos que optimice
# Agregamos uno de ellos al modelo (ctrl2)
mrM2.vIB<-update(mrM2, control= ctrl2)
mrM2.vIB <- update(mrM2.vIB, weights=varIdent(form=~1|Uso))
# Al utilizar el mismo nombre en ambas sentencias, sobre-escribo el modelo.
```

```
mrM2.vF <- update(mrM2, weights=varFixed(~ Prof))
mrM2.vP <- update(mrM2, weights=varPower(form=~ Prof))
mrM2.vE <- update(mrM2, weights=varExp(form=~ Prof))
```

**# 6.b- Comparación de modelos**

```
library("bbmle")
```

```
AICtab(mrM2, mrM2.vI, mrM2.vIB, mrM2.vF, mrM2.vP, mrM2.vE,
       weights = T, delta = TRUE, base=T, sort = TRUE)
```

```
      AIC  dAIC df weight
mrM2.vE  14.8   0.0  7     1
mrM2.vP  33.3  18.6  7 <0.001
```

```

mrM2      37.3 22.6 6 <0.001
mrM2.vIB  38.2 23.4 7 <0.001
mrM2.vI   40.4 25.6 8 <0.001
mrM2.vF   56.9 42.2 6 <0.001

```

```

# El modelo con la función de la varianza exponencial para profundidad
("mrM2.vE")
# tuvo la mejor bondad de ajuste dentro del conjunto de modelos comparados.

```

#### # 6.c- Comparamos los variogramas

```

# entre el modelo inicial ("mrM0") y los mejores ("mrM2" y "mrM2.vE")
plot(Variogram(mrM0, form= ~ Prof|Bloque/Parcela, resType="normalized",
robust = T))
plot(Variogram(mrM2, form= ~ Prof|Bloque/Parcela, resType="normalized",
robust = T))
plot(Variogram(mrM2.vE, form= ~ Prof|Bloque/Parcela, resType="normalized",
robust = T))
# Observar que los valores de correlación del semivariograma en el modelo
"mrM2.vE" están alrededor de 1.

```

#### # 6.d- Verificación del mejor modelo

##### # 6.d.1- Residuales

```

Er2VE<-resid(mrM2.vE, type="normalized")
Fit2VE<-fitted(mrM2.vE)
plot(Fit2VE, Er2VE, xlab = "Predichos", ylab = "Residuos estandarizados")
abline(0,0, col="red")

plot(suelo$Prof,Er2VE, xlab = "Profundidad", ylab = "Residuos estandarizados")
abline(0,0, col="red", lwd=3)

plot(suelo$Uso,Er2VE, xlab = "Uso", ylab = "Residuos estandarizados")
plot(suelo$Parcela,Er2VE, xlab = "Parcela", ylab = "Residuos estandarizados")
plot(suelo$Bloque,Er2VE, xlab = "Bloque", ylab = "Residuos estandarizados")

```

##### # 6.d.2- Normalidad

```

qqnorm(Er2VE, main="modelo")
qqline(Er2VE)

```

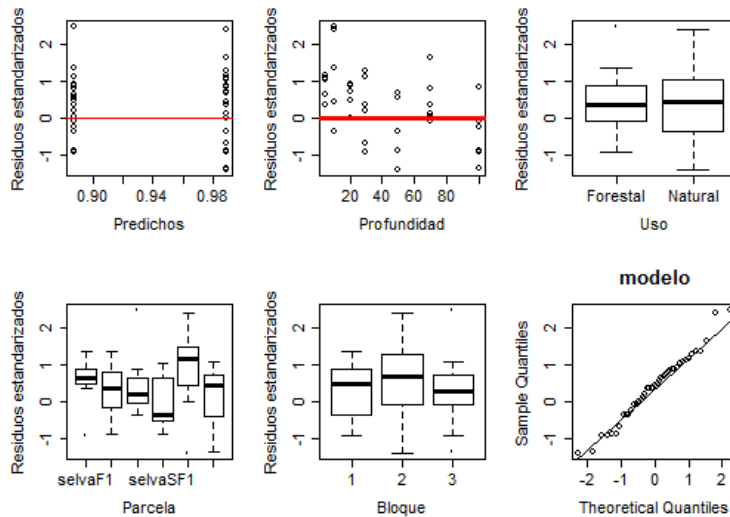


Figura 10. 4: residuales normalizados del modelo "mrM2.vE" en relación a los ajustados por el modelo, los predictores: profundidad, uso, parcela y bloque. Gráfico de normalidad (cuantiles teóricos) para el mismo modelo.

## # 7- Ajustamos el componente fijo del modelo

# Utilizaremos como marco inferencial el ANOVA

```
anova(mrM2.vE)
numDF denDF F-value p-value
(Intercept) 1 36 324.8551 <.0001
Uso 1 2 0.9789 0.4267
```

# El modelo final

```
mrM2.vE$call
lme.formula(fixed = func, data = suelo, random = ~1 | Bloque/Parcela
, correlation = corSpher(form = ~Prof | Bloque/Parcela), weights = v
arExp(form = ~Prof))
> func N ~ Uso
```

# El modelo se puede representar de manera matemática como:

$$Y_i \sim N(\mu_i, \sigma_{\epsilon, profundidad}^2)$$

$$\mu_i = \beta_{0j[i]} + \beta_1 * Uso_i$$

$$\beta_{0j} \sim N(\mu_{\beta_{0j}}, \sigma_{\beta_0}^2)$$

$$\mu_{\beta_{0j}} = \alpha_{0[k]}$$

$$\alpha_{0k} \sim N(\mu_{\alpha_{0k}}, \sigma_{\alpha_0}^2)$$

$i = 1, 2, \dots, 42$  unidad de suelo

$j = 1, 2, \dots, 6$  parcelas

$k = 1, 2, 3$  bloques

**Donde,**

$Y_i$  = cantidad nitrógeno en la unidad de suelo

Coefficientes:

$\beta_{0j}$  = intercepción

$\beta_1 = \text{uso}$   
 $\alpha_{0j} = \text{intercepción}$   
 $\sigma_{\alpha 0}^2 = \text{varianza entre bloques}$   
 $\sigma_{\beta 0}^2 = \text{varianza entre parcelas}$   
 $\sigma_{\epsilon}^2 = \text{varianza residual}$

## # 8- Modelos alternativos

### # 8.a- Generar modelos alternativos

# ¿Qué hubiera pasado con la correlación si la profundidad  
 # hubiera estado incluida como variable predictora en los  
 # componentes fijos del modelo?  
 suelo\$Prof\_f<-as.factor(suelo\$Prof)  
 Alt<- lme(N~ Uso+Prof\_f,random=~1|Bloque/Parcela, data = suelo)  
 # En este caso se incorpora profundidad como factor a fines didácticos.  
 # Pero profundidad podría haber sido incorporada como variable cuantitativa.

# **Tarea:** escribir el modelo con profundidad como variable continua.  
 # Discutir ventajas y desventajas de ambas aproximaciones.

```
plot(Alt)
```

# Observamos los variogramas del modelo inicial (mrM0), el corregido  
 (mrM2.VE) y el alternativo (Alt).  
 plot(Variogram(mrM0, form= ~ Prof|Bloque/Parcela, resType="normalized",  
 robust = T))

```
plot(Variogram(mrM2.vE, form= ~ Prof|Bloque/Parcela, resType="normalized",  
robust = T))
```

```
plot(Variogram(Alt, form= ~ Prof|Bloque/Parcela, resType="normalized", robust  
= T))
```

# TAREA: Comparar y discutir los variogramas de los tres modelos.

### # 8.b- Inferencia

```
anova(Alt)
```

	numDF	denDF	F-value	p-value
(Intercept)	1	30	533.9006	<.0001
Uso	1	2	8.5242	0.1
Prof_f	6	30	26.4053	<.0001

# El modelo sugiere que el contenido de N varía con la profundidad  
 # Relacionar con lo encontrado en los modelos anteriores.

### # 8.c- Test de comparaciones múltiples

```
library(multcomp)
summary(glht(Alt, linfct=mcp(Prof_f="Tukey")))
```

```

Simultaneous Tests for General Linear Hypotheses
Multiple Comparisons of Means: Tukey Contrasts
Fit: lme.formula(fixed = N ~ Uso + Prof_f, data = suelo, random =
~1 | Bloque/Parcela)

```

## Linear Hypotheses:

	Estimate	Std. Error	z value	Pr(> z )	
10 - 5 == 0	0.16649	0.12404	1.342	0.83203	
20 - 5 == 0	0.05197	0.12404	0.419	0.99959	
30 - 5 == 0	-0.15874	0.12404	-1.280	0.86143	
50 - 5 == 0	-0.59925	0.12404	-4.831	< 0.001	***
70 - 5 == 0	-0.75221	0.12404	-6.064	< 0.001	***
100 - 5 == 0	-0.99150	0.12404	-7.993	< 0.001	***
20 - 10 == 0	-0.11452	0.12404	-0.923	0.96896	
30 - 10 == 0	-0.32523	0.12404	-2.622	0.11935	
50 - 10 == 0	-0.76574	0.12404	-6.173	< 0.001	***
70 - 10 == 0	-0.91870	0.12404	-7.406	< 0.001	***
100 - 10 == 0	-1.15799	0.12404	-9.335	< 0.001	***
30 - 20 == 0	-0.21071	0.12404	-1.699	0.61690	
50 - 20 == 0	-0.65122	0.12404	-5.250	< 0.001	***
70 - 20 == 0	-0.80418	0.12404	-6.483	< 0.001	***
100 - 20 == 0	-1.04347	0.12404	-8.412	< 0.001	***
50 - 30 == 0	-0.44051	0.12404	-3.551	0.00692	**
70 - 30 == 0	-0.59347	0.12404	-4.784	< 0.001	***
100 - 30 == 0	-0.83276	0.12404	-6.714	< 0.001	***
70 - 50 == 0	-0.15296	0.12404	-1.233	0.88139	
100 - 50 == 0	-0.39225	0.12404	-3.162	0.02626	*
100 - 70 == 0	-0.23929	0.12404	-1.929	0.46064	

---  
 Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
 (Adjusted p values reported -- single-step method)

# El Test compara las profundidades de a pares bajo la hipótesis nula de que no hay diferencias entre ellas.

# En este caso, por ejemplo, el test indica que el contenido de N es similar hasta los 20 cm de profundidad (no hay diferencias entre 5 y 10, 5 y 20, 5 y 30, 10 y 20, 10 y 30, 20 y 30).

# En cambio hay diferencias entre la profundidad 5 y las profundidades mayores de 50 (5 y 50, 5 y 70, 5 y 100).

# Un análisis similar pueden hacer para las profundidades de 10 y 20.

### # 9- Gráficos del modelo final

```
layout(matrix(1:1,1,1))
medias=with(suelo,tapply(N,Prof_f,mean))
medias
desvios=with(suelo,tapply(N,Prof_f,sd))
desvios
fig= barplot(medias, ylim=c(0,3), xlab="Profundidad (cm)",ylab= "Nitrógeno (g.Kg-1)")
arrows(fig, medias+desvios, fig, medias-desvios, angle=90,code=3)
# ATENCIÓN: estamos graficando el desvío estándar y no el error estándar.
```

```
summary(Alt)$tTable
```

# De la tabla anterior extraemos los valores de Error estándar

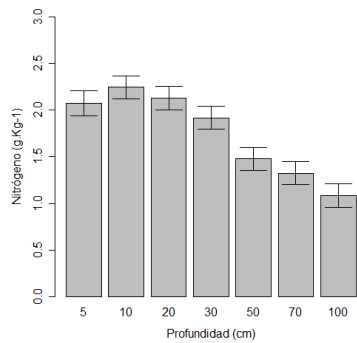
```
summary(Alt)$tTable[-c(2),2]
```

```
std<-summary(Alt)$tTable[-c(2),2]
```

```
fig= barplot(medias, ylim=c(0,3), xlab="Profundidad (cm)",ylab= "Nitrógeno (g.Kg-1)")
```

```
arrows(fig,medias+std,fig,medias-std, angle=90,code=3)
```





**Figura 10. 5: valores de nitrógeno del suelo (g. kg<sup>-1</sup>suelo) para las distintas profundidades. Las barras muestran el valor medio  $\pm$  error estándar.**

# **Tarea:** hacer un gráfico final para el modelo que considera "profundidad como variable continua"

# (pueden basarse en gráficos presentados en prácticas anteriores)

### # 10- Resumen

# En este caso, trabajamos sobre un caso especial de correlación espacial: medidas a intervalos desiguales.

# Generamos variogramas para evaluar la correlación.

# Incorporamos al modelo estructuras de correlación espacial.

# Incorporamos funciones de la varianza para corregir problemas de Homocedasticidad (heterogeneidad en las varianzas).

# Verificamos la validez del mejor modelo a partir del gráfico de variograma y, de la distribución de los residuales estandarizados.

# Utilizamos como marco inferencial frecuentista: anova(modelo) y, comparaciones múltiples a posteriori (Tukey para modelos mixtos)

# Propusimos modelos alternativos con profundidad como predictor

# Graficamos el modelo final

### # 11- Series de tiempo

# Para los interesados en este tipo de análisis, R tiene infinitas posibilidades de análisis de series de tiempo

# Les dejamos un enlace que describe estas capacidades:

# <http://cran.r-project.org/web/views/TimeSeries.html>

### # 12- Agradecimiento

# Datos gentilmente provistos por Paola Roxana Ecclesia y Gervasio Piñeiro de la Facultad de Agronomía de la Universidad de Buenos Aires. Ecclesia, R., E. G. Jobbagy, R. B. Jackson, F. Biganzoli y G. Piñeiro. 2012. Shifts in soil organic carbon for plantation and pasture establishment in native forests and grasslands of South America. *Global Change Biology* 18(10): 3237-3251. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.

# Modelos con jerarquías en experimentos observacionales

---

## Contenidos

<b># CAPÍTULO 11: Modelos con jerarquías en experimentos observacionales....</b>	<b>171</b>
# 1- Configuración inicial.....	171
# 2- El caso: Productividad de ciprés nativo.....	171
# 3- Explorar los datos.....	172
# 4- Multicolinealidad.....	172
# 5- Modelo global.....	173
# 6- Funciones de la varianza.....	179
# 7- Inferencia multimodelo.....	181
# 8- Modelo estandarizado.....	184
# 9- Verificar el mejor modelo.....	185
# 10- Presentamos el modelo.....	188
# 11- Graficar el modelo final.....	189
# 12- Resumen.....	192
# 13- Agradecimiento.....	192

---



---

## # CAPÍTULO 11: Modelos con jerarquías en experimentos observacionales

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo

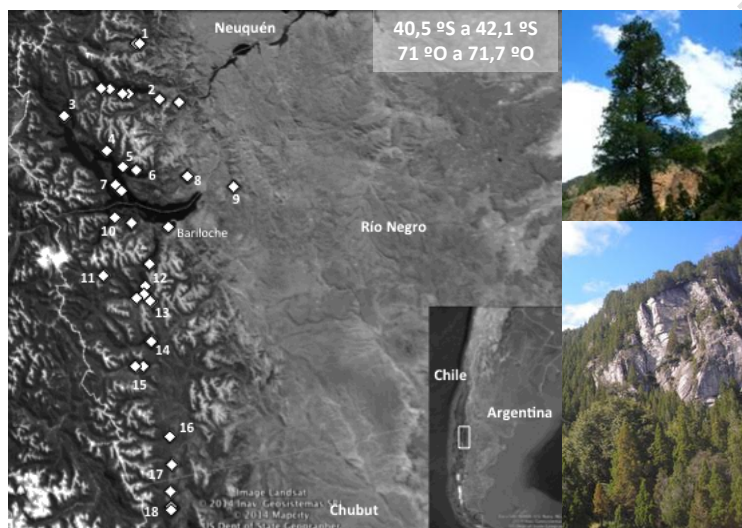
```
# Utilizar alguna de las maneras vistas en las prácticas anteriores
```

```
# Datos <- read.csv("11_Cipres.csv")
summary(Datos)
```

#### # 1.c- Acondicionar la tabla de datos

```
# Excluimos los datos faltantes ("NAs") de la tabla
cipres<-na.omit(Datos)
```

### # 2- El caso: Productividad de ciprés nativo



Estimar la productividad de sitio es imprescindible para la actividad forestal dado que determina en parte, el volumen potencial de madera. El ciprés de la cordillera es una especie nativa de amplia distribución en la Patagonia y de gran importancia económica con un desarrollo técnico-productivo

incipiente.

El objetivo es generar un modelo capaz de clasificar sitios según su productividad para ciprés de la cordillera (*Austrocedrus chilensis*) considerando características topográficas, climáticas y propias del rodal.

#### # 2.a- Pregunta de interés

```
# ¿Qué características ambientales (climáticas, topográficas,
# poblacionales)
# determinan el crecimiento del ciprés nativo?
```

#### # 2.b- Variable respuesta

```
# Volumen de ciprés (m3)
```

#### # 2.c- Variable/s predictoras de interés

```
# # Ubicación:
# Latitud
# Longitud
```

```

#
# # Topográficas:
# Orient_N (sin unidad)
# Orient_E (sin unidad)
# Altitud (en msnm)
# Pendiente (en grados)
#
# # Climáticas:
# Precipitacion (mm/año)
# Temp_ver (°C promedio de los meses de verano: enero, febrero, marzo)
# Temp_inv (°C promedio de los meses de invierno: julio, agosto, septiembre)
#
# # Poblacionales:
# Densidad_cipres (n/ha)
# Edad (años)

```

### # 2.d- Identificar las escalas y las variables asociadas

```

# Individuos de cipres en parcelas.
# Los individuos estan identificados en "N_tarugos". A nivel de individuo
# se midió la "Edad" y el "Vol" (volumen) de los cipreses
# Las parcelas estan agrupadas en regiones.
# A nivel de parcela se determinó la densidad de cipreses ("Densidad_cipres"),
# las temperaturas medias ("Temp_ver" y "Temp_inv"), la precipitación,
# la pendiente, la altitud,
# la Orientación Norte y la orientación Este, la Latitud y la Longitud.
# No hubo variables medidas a nivel de Región.

```

### # 3- Explorar los datos

```

with(cipres, dotchart(Vol, xlab="Volumen (m3)", ylab=" Datos (# de orden)"))
with(cipres, boxplot(Vol~Region, ylab="Volumen (m3)", xlab="Región"))
with(cipres, plot(Vol~Edad, ylab="Volumen (m3)", xlab="Edad"))

```

# **TAREA:** pueden generar otros gráficos usando como guía las prácticas anteriores.

### # 4- Multicolínealidad

```

panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y, method= "pearson", use= "pairwise.complete.obs"))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

```

```

panel.linea = function(x, y, ...) {
  tmp <- lm(y ~ x, na.action = na.omit)
  abline(tmp)
  points(x, y) }

```

```

pairs(cipres[,c("Altitud", "Orient_N", "Orient_E",
              "Pendiente", "Precipitacion", "Temp_ver", "Temp_inv",
              "Densidad_cipres", "Edad")],
      lower.panel = panel.cor,
      upper.panel=panel.linea,
      cex.labels=1,
      labels=c("Altitud", "Orient_N", "Orient_E",
              "Pendiente", "Precipitacion", "Temp_ver", "Temp_inv",
              "Densidad_cipres", "Edad"))
    
```

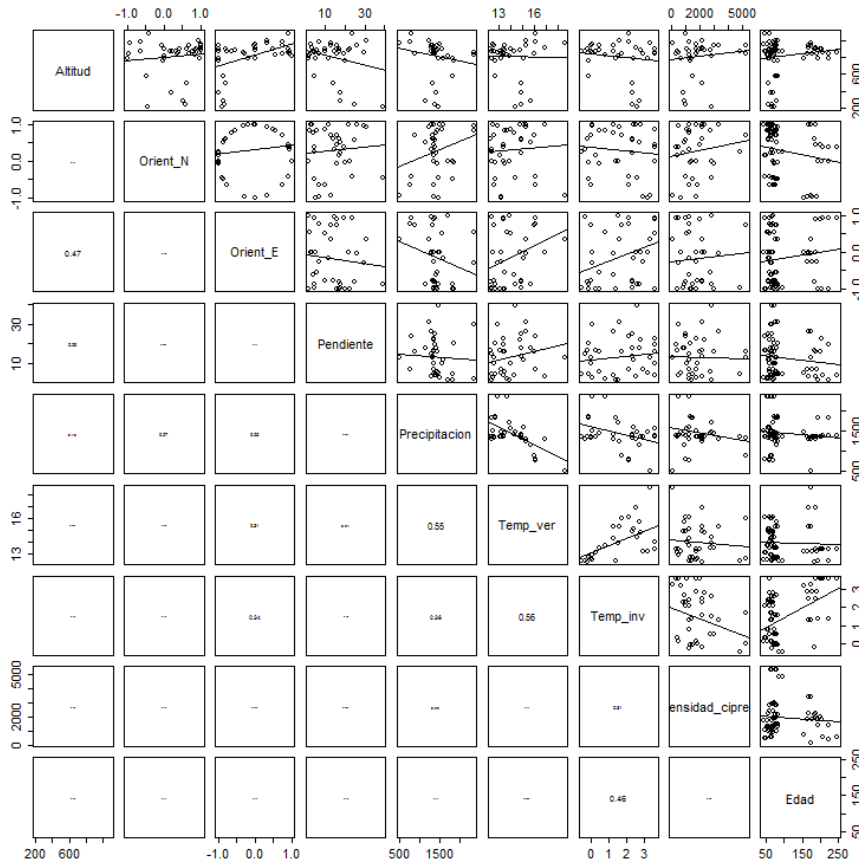


Figura 11 1: relación entre variable cuantitativas: altitud, orientación Norte, orientación Este, pendiente, precipitación, temperatura media de verano, temperatura media de invierno, densidad de árboles de ciprés, edad promedio de los árboles de ciprés. Distribución de los puntos en el triángulo superior de la matriz y valor de coeficiente de Pearson en el triángulo inferior de la matriz.

# Dada la relación no lineal entre Orientación Norte y Orientación E  
 # ("Orient\_N" y "Orient\_E"), utilizaremos "Orient\_N"  
 # como variable predictora.

### # 5- Modelo global

# **IMPORTANTE** : Este modelo debería tener los parámetros,  
 # todos los efectos potencialmente relevantes  
 # y, describir los mecanismos causales basados en el conocimiento previo.  
 # Es decir, el modelo debería representar nuestra hipótesis.

### # 5.a- Ajustar la componente aleatoria

# Generamos modelos con la misma componente fija  
 # y distintas estructuras aleatorias (en este caso, pendientes)  
 # Luego, comparamos los modelos según su AIC.  
 # Los modelos son ajustados por el método de REML

# En este caso, excluimos las interacciones por cuestiones didácticas (funciones que utilizaremos más adelante demorarían demasiado para un práctico).

```
library(lme4)
modelGlobA<-lmer(Vol ~ Orient_N + Pendiente + Altitud + Temp_ver +
  Temp_inv + Precipitacion + Densidad_cipres + Edad +
    (1|Region/Parcela), data= cipres, REML=TRUE)
```

# Puede que lean el siguiente mensaje:  
 # Warning message:  
 # Some predictor variables are on very different scales: consider rescaling  
 # Más adelante veremos cómo re-escalar los datos.

```
modelGlobB<-lmer(Vol ~ Orient_N + Pendiente + Altitud + Temp_ver +
  Temp_inv + Precipitacion + Densidad_cipres + Edad +
    (1|Region/Parcela) + (Temp_ver|Region),
  data = cipres,
  REML=TRUE)
```

```
modelGlobC<-lmer(Vol ~ Orient_N + Pendiente + Altitud + Temp_ver +
  Temp_inv + Precipitacion + Densidad_cipres + Edad +
    (1|Region/Parcela) + (Temp_inv|Region),
  data = cipres,
  REML=TRUE)
```

```
modelGlobD<-lmer(Vol ~ Orient_N + Pendiente + Altitud + Temp_ver +
  Temp_inv + Precipitacion + Densidad_cipres + Edad +
    (1|Region/Parcela) + (Altitud|Region),
  data = cipres,
  REML=TRUE)
```

### # 5.b- Comparar los modelos

```
library("bbmle")
AICtab(modelGlobA, modelGlobB, modelGlobC, modelGlobD,
  weights = T, delta = TRUE, base=T, sort = TRUE)
```

	AIC	dAIC	df	weight
modelGlobA	152.9	0.0	12	0.630
modelGlobC	154.4	1.5	15	0.303
modelGlobD	158.7	5.8	15	0.035
modelGlobB	158.9	6.0	15	0.032

# El modelo de mejor bondad de ajuste entre los modelos especificados es el modelo "modelGlobA".

### # 5.c- Verificación del modelo

# Debemos hacer una verificación a priori del modelo:

```
# - Suficientes grados de libertad,
# - verificar la estimación de los parámetros y
# - el ajuste del modelo,
# - la distribución de los residuales, etc.
summary(modelGlobA)
```

### # 5.c.1- Ajustados vs. los residuales del modelo

```
EGlobA<-resid(modelGlobA, type="pearson")
FGlobA<-fitted(modelGlobA)
plot(x=FGlobA, y=EGlobA, xlab="Ajustados", ylab="Residuales estandarizados")
abline(a=0,b=0, col="violet", lw=2)
```

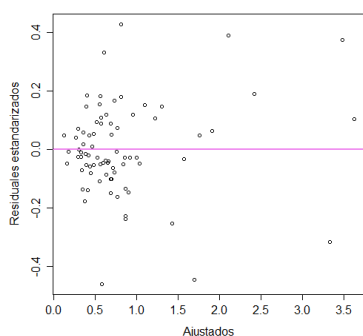


Figura 11 2: residuales en relación a los valores ajustados por el modelo "modelGlobA"

### # 5.c.2- Residuales en función de las variables

```
layout(matrix(1:6, 2,3, byrow = TRUE))
# sentencia que organiza la ventana gráfica
```

```
plot(EGlobA~cipres$Region, main="Region", xlab="Region", ylab="Residuos normalizados")
```

```
plot(EGlobA~cipres$Orient_N, main="Orient_N", xlab="Orient_N", ylab="Residuos normalizados")
```

```
abline(a=0,b=0, col="violet", lw=2)
```

```
plot(EGlobA~cipres$Pendiente, main="Pendiente", xlab="Pendiente", ylab="Residuos normalizados")
```

```
abline(a=0,b=0, col="violet", lw=2)
```

```
plot(EGlobA~cipres$Altitud, main="Altitud", xlab="Altitud", ylab="Residuos normalizados de M.lme1")
```

```
abline(a=0,b=0, col="violet", lw=2)
```

```
plot(EGlobA~cipres$Temp_ver, main="Temp_ver", xlab="Temp_ver", ylab="Residuos normalizados de M.lme1")
```

```
abline(a=0,b=0, col="violet", lw=2)
```

```
plot(EGlobA~cipres$Temp_inv, main="Temp_inv", xlab="Temp_inv", ylab="Residuos normalizados de M.lme1")
```

```
abline(a=0,b=0, col="violet", lw=2)
```

```
plot(EGlobA~cipres$Precipitacion, main="Precipitacion", xlab="Precipitacion", ylab="Residuos normalizados de M.lme1")
```

```
abline(a=0,b=0, col="violet", lw=2)
```

```
plot(EGlobA~cipres$Densidad_cipres, main="Densidad_cipres",
     xlab="Densidad_cipres", ylab="Residuos normalizados de M.lme1")
abline(a=0,b=0, col="violet", lw=2)
plot(EGlobA~cipres$Edad, main="Edad", xlab="Edad", ylab="Residuos
normalizados de M.lme1")
abline(a=0,b=0, col="violet", lw=2)
```

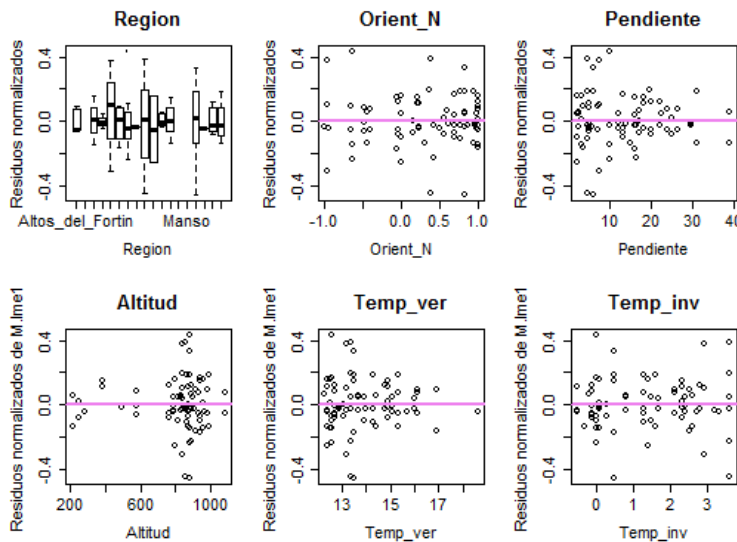


Figura 11 3: residuales del modelo "modelGloBA" en relación a las variables predictoras: región, orientación Norte, pendiente, altitud, temperatura media de verano y temperatura media de invierno.

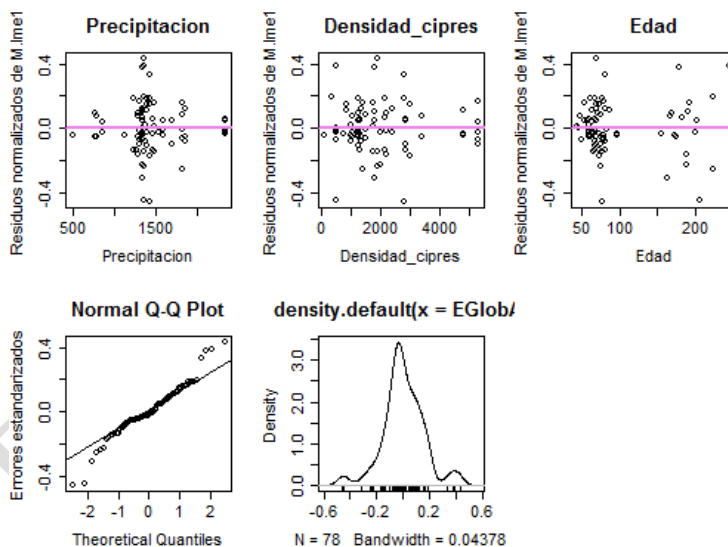


Figura 11 4: residuales del modelo "modelGloBA" en relación a las variables predictoras: precipitación, densidad y edad promedio de árboles de ciprés. Gráfico de normalidad en la fila inferior.

# La distribución de los residuos presenta algún patrón respecto de los predictores "Pendiente", "Altitud", "Temp\_ver", "Precipitación", "Densidad\_cipres" y "Edad"

#### # 5.c.4- Normalidad

```
qqnorm(EGlobA, ylab="Errores estandarizados")
qqline(EGlobA)
```



```
plot(density(EGlobA))
rug(EGlobA)
```

### # 5.c.5- Distribución espacial de los residuos

```
library("sp")
```

```
mydata<- data.frame(EGlobA, cipres$Longitud, cipres$Latitud)
colnames(mydata) <- c("EGlobA", "Longitud", "Latitud")
coordinates(mydata) <- ~Longitud+Latitud
bubble(mydata, "EGlobA", col = c("black","red"),
       main = "Residuales", xlab = "Longitud",
       ylab = "Latitud")
```

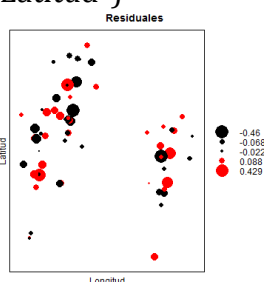


Figura 11 5: distribución espacial de los residuos normalizados del modelo "modelGlobA". Los puntos rojos indican valores positivos y los negro indican valores negativos. El tamaño de los puntos indica la magnitud del residual.

# Los puntos negros en el gráfico indican valores negativos y los puntos rojos, positivos.  
 # Una distribución deseable de los residuos no debería mostrar patrones de distribución espacial (Ej.: valores positivos más importantes ó, predominancia de algún sector)

### # 5.c.6- Variogramas

# **IMPORTANTE:** para agregar correlaciones debemos escribir el modelo bajo la sintaxis de la función "lme" del paquete "nlme"

```
library(nlme)
GlobAlme<-lme(Vol ~ Orient_N + Pendiente + Altitud + Temp_ver + Temp_inv +
  Precipitacion + Densidad_cipres + Edad,
  random=~1|Region/Parcela, data= cipres, method="REML")
```

```
EGlob<-resid(GlobAlme, type="normalized")
```

### # TAREA:

# - generar el gráfico de burbujas para los residuos de este modelo y comparar con el anterior

```
library(gstat)
```

```
coord<- data.frame(cipres$Longitud, cipres$Latitud)
colnames(coord) <- c("Longitud", "Latitud")
coordinates(coord) <- ~Longitud+Latitud
```

```
Vario <- variogram(EGlob ~ 1,coord)
plot(Vario)
```

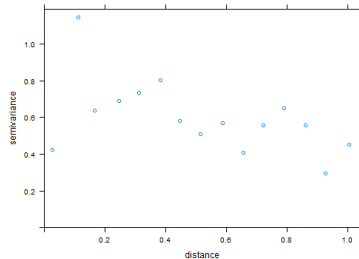


Figura 11 6: semivariograma de los residuales normalizados del modelo "GlobAlme"

# Generamos modelos con estructuras de correlación con el objetivo de corregir el patrón de distribución espacial de los residuales que describimos anteriormente.

```
GlobAlmeLin<-update(GlobAlme,correlation=corLin(form = ~
Longitud+Latitud))
GlobAlmeExp<-update(GlobAlme,correlation=corExp(form = ~
Longitud+Latitud))
GlobAlmeSpher<-update(GlobAlme,correlation=corSpher(form = ~
Longitud+Latitud))
GlobAlmeGaus<-update(GlobAlme,correlation=corGaus(form = ~
Longitud+Latitud))
GlobAlmeRatio<-update(GlobAlme,correlation=corRatio(form = ~
Longitud+Latitud))
```

# Comparamos los modelos

```
AICtab(GlobAlme,GlobAlmeLin,GlobAlmeExp,GlobAlmeSpher,GlobAlmeGaus,GlobAlmeRatio, weights=T, sort=T, base=T)
```

```
#           AIC    dAIC  df weight
# GlobAlme    152.9    0.0  12  0.35
# GlobAlmeLin  154.9    2.0  13  0.13
# GlobAlmeSpher 154.9    2.0  13  0.13
# GlobAlmeGaus  154.9    2.0  13  0.13
# GlobAlmeExp  154.9    2.0  13  0.13
# GlobAlmeRatio 154.9    2.0  13  0.13
```

# A pesar de ser el modelo sin estructura de correlación el mejor, la diferencia respecto de los otros, es de 2 AIC.

# Dado que la diferencia en el ajuste (AIC) es pequeña, generamos gráficos para observar y comparar la distribución de los residuales.

```
EGlobAlmeLin<-resid(GlobAlmeLin, type="normalized")
```

```
mydata<- data.frame(EGlobAlmeLin, cipres$Longitud, cipres$Latitud)
colnames(mydata) <- c("EGlobAlmeLin", "Longitud", "Latitud")
coordinates(mydata) <- ~Longitud+Latitud
bubble(mydata, "EGlobAlmeLin", col = c("black","red"),
       main = "Residuales", xlab = "Longitud",
       ylab = "Latitud")
```

# En este caso, volvemos a observar un patrón en la distribución de los puntos dado por errores pequeños y grandes que se superponen.

#### # TAREA:

# a- Generar gráficos de burbujas para observar la distribución de los residuales en los otros modelos.

# b- Discutir: ¿El modelo es adecuado?

# Volveremos a revisar el variograma más adelante.

## # 6- Funciones de la varianza

### # 6.a- Generar modelos

# Con variables continuas se pueden utilizar las siguientes funciones de la varianza:

```
modelVf1<- update(GlobAlme, weights = varFixed(~ Pendiente))
modelVf2<- update(GlobAlme, weights = varFixed(~ Altitud))
modelVf3<- update(GlobAlme, weights = varFixed(~ Temp_ver))
modelVf4<- update(GlobAlme, weights = varFixed(~ Precipitacion))
modelVf5<- update(GlobAlme, weights = varFixed(~ Densidad_cipres))
modelVf6<- update(GlobAlme, weights = varFixed(~ Edad))
```

```
modelVp1<- update(GlobAlme, weights = varPower(form=~ Pendiente))
modelVp2<- update(GlobAlme, weights = varPower(form=~ Altitud))
modelVp3<- update(GlobAlme, weights = varPower(form=~ Temp_ver))
modelVp4<- update(GlobAlme, weights = varPower(form=~ Precipitacion))
modelVp5<- update(GlobAlme, weights = varPower(form=~ Densidad_cipres))
modelVp6<- update(GlobAlme, weights = varPower(form=~ Edad))
```

```
modelVe1<- update(GlobAlme, weights = varExp(form=~ Pendiente))
modelVe2<- update(GlobAlme, weights = varExp(form=~ Altitud))
modelVe3<- update(GlobAlme, weights = varExp(form=~ Temp_ver))
modelVe4<- update(GlobAlme, weights = varExp(form=~ Edad))
```

# Es posible combinar las funciones

```
modelVp4_Ve4<-update(GlobAlme, weights = varComb(varPower(form=~
Precipitacion), varExp(form=~ Edad)))
```

### # 6.b- Comparar modelos

```
library("bbmle")
AICtab(GlobAlme, modelVf1, modelVf2, modelVf3,modelVf4, modelVf5,
       modelVf6,
```

```

modelVp1, modelVp2, modelVp3, modelVp4, modelVp5, modelVp6,
modelVe1, modelVe2, modelVe3, modelVe4,
modelVp4_Ve4,
weights = T, delta = TRUE, base=T, sort = TRUE)

```

	AIC	dAIC	df	weight
modelVp6	140.6	0.0	13	0.4983
modelVe4	142.6	2.0	13	0.1846
modelVf6	142.7	2.1	12	0.1723
modelVp4_Ve4	143.4	2.8	14	0.1203
modelVe1	149.6	9.0	13	0.0055
modelVe3	150.1	9.5	13	0.0043
modelVp3	150.3	9.7	13	0.0039
modelVf2	150.9	10.3	12	0.0028
modelVp1	151.1	10.5	13	0.0026
modelVp2	152.9	12.3	13	0.0011
modelVe2	152.9	12.3	13	0.0011
GlobAlme	152.9	12.3	12	0.0010
modelVp4	154.0	13.4	13	<0.001
modelVf4	154.3	13.7	12	<0.001
modelVp5	154.3	13.7	13	<0.001
modelVf3	154.3	13.8	12	<0.001
modelVf5	164.6	24.0	12	<0.001
modelVf1	169.7	29.1	12	<0.001

```

# El modelo de mejor ajuste es el modelo "modelVp6"
# con las funciones de la varianza Power para Edad
# entre el conjunto de modelos comparados y tiene un delta AIC de 26
# respecto del segundo mejor modelo y, de 12.3 respecto del modelo sin función
de la varianza (GlobAlme).

```

### # 6.c- Verificar del modelo con funciones de la varianza

#### # 6.c.1- Ajustados vs. los residuales del modelo

```

# Comparamos los modelos sin y con funciones de las varianzas

```

```

layout(matrix(1:2, 1, 2))
EGloblme<-resid(GlobAlme, type="normalized")
FGloblme<-fitted(GlobAlme)
plot(x=FGloblme, y=EGloblme, xlab="Ajustados", ylab="Residuales
estandarizados", main="Modelo inicial")
abline(a=0,b=0, col="violet", lw=2)

```

```

EGlob<-resid(modelVp6, type="normalized")
FGlob<-fitted(modelVp6)
plot(x=FGlob, y=EGlob, xlab="Ajustados", ylab="Residuales estandarizados",
main="Modelo con varFunc")
abline(a=0,b=0, col="violet", lw=2)

```

#### # 6.c.2- Residuales en función de las variables

```

layout (matrix(1:6, 2,3, byrow = TRUE)) # sentencia que organiza la ventana
gráfica en cuatro cuadrantes

```

```

plot(EGlob~cipres$Region, main="Region", xlab="Region", ylab="Residuos
normalizados")

```

```

plot(EGlob~cipres$Orient_N, main="Orient_N", xlab="Orient_N", ylab="Residuos
normalizados")
abline(a=0,b=0, col="violet", lw=2)
plot(EGlob~cipres$Pendiente, main="Pendiente", xlab="Pendiente",
ylab="Residuos normalizados")
abline(a=0,b=0, col="violet", lw=2)
plot(EGlob~cipres$Altitud, main="Altitud", xlab="Altitud", ylab="Residuos
normalizados de M.lme1")
abline(a=0,b=0, col="violet", lw=2)
plot(EGlob~cipres$Temp_ver, main="Temp_ver", xlab="Temp_ver",
ylab="Residuos normalizados de M.lme1")
abline(a=0,b=0, col="violet", lw=2)
plot(EGlob~cipres$Temp_inv, main="Temp_inv", xlab="Temp_inv",
ylab="Residuos normalizados de M.lme1")
abline(a=0,b=0, col="violet", lw=2)
plot(EGlob~cipres$Precipitacion, main="Precipitacion", xlab="Precipitacion",
ylab="Residuos normalizados de M.lme1")
abline(a=0,b=0, col="violet", lw=2)
plot(EGlob~cipres$Densidad_cipres, main="Densidad_cipres",
xlab="Densidad_cipres", ylab="Residuos normalizados de M.lme1")
abline(a=0,b=0, col="violet", lw=2)
plot(EGlob~cipres$Edad, main="Edad", xlab="Edad", ylab="Residuos
normalizados de M.lme1")
abline(a=0,b=0, col="violet", lw=2)

```

### # 6.c.3- Normalidad

```

qqnorm(EGlob, ylab="Errores estandarizados")
qqline(EGlob)

```

```

plot(density(EGlob))
rug(EGlob)

```

### # 6.c.4- Variograma

```

# Variograma inicial
plot(Vario)

```

```

# Variograma Vp
VarioVp <- variogram(EGlob ~ 1, coord)
plot(VarioVp)

```

# **TAREA:** Discutir: ¿El modelo es adecuado?

### # 7- Inferencia multimodelo

```

# Utilizaremos la función "dedge()" en el paquete MuMIn para
# comparar todos los modelos de efectos fijos posibles a partir del
# modelo global ajustado
# Luego compararemos los modelos generados
# con el AIC mediante la función "model.sel".
# Dado que compararemos modelos con distinta componente fija
# (pero la misma estructura aleatoria) utilizaremos el método de ML

```

```
# para estimar los modelos.  
modelML<-update(modelVp6, method="ML")
```

### # 7.a- Generamos todos los modelos posibles

```
library(MuMIn)
```

```
modelGen<-dredge(modelML)  
# Este proceso puede demorar unos minutos dado que debe estimar gran  
cantidad de modelos.
```

```
dim(modelGen)
```

```
[1] 253 14
```

```
# Observar, la función "dredge" generó 253 modelos (hubo 3 que no  
convergió) a  
# partir de los predictores incluidos en el modelo global.
```

### # 7.b- Ordenar los modelos según su bondad de ajuste

```
# La función "model.sel" ordena los modelos según el ajuste  
# (AIC, BIC, AICc, etc).  
# Además, genera una ponderación del ajuste relativa a los  
# modelos incluidos ("weight").  
# Por defecto ordena los modelos según su AICc, un índice de AIC  
# que corrige cuando el número de observaciones es bajo.  
# El argumento "rank" permite cambiar el criterio de ajuste  
RankAIC <- model.sel(modelGen, rank="AIC")  
RankAIC
```

```
# Guardamos el resultado de los 256 modelos en una planilla de cálculos.  
write.table(RankAIC, file = "12_Cipres_Ranking de Modelos.csv", append = F, sep  
= "\t")
```

```
# IMPORTANTE: al abrir la planilla de cálculos que acabamos de generar,  
corregir la primera fila.
```

```
# Deben correr un lugar a la derecha las columnas.
```

```
# La primera columna identifica el número del modelo
```

```
# y la segunda la el valor estimado de las intercepciones.
```

```
# Las siguientes columnas muestran los valores estimados para los siguientes  
predictores.
```

```
# Cuando el predictor no estuvo incluido en el modelo, aparece un NA en la celda.
```

```
# Las últimas cinco columnas muestran:
```

```
#- la cantidad de parámetros del modelo (df)
```

```
#- el log de la máxima verosimilitud (logLik)
```

```
#- el AIC
```

```
#- el delta
```

```
#- el weight
```

```
# Podemos tomar un subconjunto de los mejores modelos
```

```
RankAIC_Mejores<-get.models(RankAIC,subset = delta <= 10)
```

```
length(names(RankAIC_Mejores))
```

```
[1] 102
```

```
# Dentro del conjunto de modelos hay 102 que se encuentran dentro
# del rango de un delta AIC menor o igual a 10
```

```
head(RankAIC)
Global model call: lme.formula(fixed = Vol ~ Orient_N + Pendiente +
  Altitud + Temp_ver +
  Temp_inv + Precipitacion + Densidad_cipres + Edad, data = cipres
,
  random = ~1 | Region/Parcela, weights = varPower(form = ~Edad),
  method = "ML")
---
```

```
Model selection table
```

	Int	Alt	Edd	Orn_N	Pnd	Prc	Temp_       inv	Temp_       ver	df	logLik	AIC	delta	w
197	2.06		6.90E				0.18	-0.15	8	-17.82	51.7	0	0.258
205	2.23		6.44E	-0.11			0.19	-0.16	9	-16.97	52	0.3	0.222
213	2.03		6.82E		0		0.18	-0.14	9	-17.40	52.8	1.15	0.145
229	2.64		6.73E			0	0.19	-0.18	9	-17.50	53	1.35	0.131
221	2.20		6.37E	-0.11	0		0.19	-0.15	10	-16.53	53.1	1.42	0.127
198	2.02	0	6.78E				0.19	-0.16	9	-17.6	53.2	1.58	0.117

```
Random terms (all models):
'1 | Region', '1 | Parcela %in% Region'
```

```
# Observar que los mejores modelos siempre incluyen a los predictores
# Edad, Temperatura de invierno (Temp_inv) y Temperatura de verano
(Temp_ver)
```

```
importance(RankAIC)
```

	Edad	Temp_inv	Temp_ver	Orient_N	Pend	Precip	Altitud
	0.998	0.891	0.862	0.449	0.359	0.307	0.302

```
Densidad_cipres
0.277
attr(,"n.models")
```

```
# Muestra la sumatoria de los "weights" de los modelos en los que aparece el
predictor.
```

```
# Es decir, si el valor de "importance" es 1 el predictor aparece en todos los
modelos generados.
```

```
# Muestra la cantidad de modelos en los que aparece un predictor.
```

```
# Observar que predictores que aparecen en la misma cantidad de modelos,
poseen distinta "importancia" indicando que aparecen en modelo de mayor o
menor "weight"
```

### # 7.c- Identificar el mejor modelo

```
AIC_BEst<-get.models(RankAIC,subset = delta <= 0)
```

```
AIC_BEst
```

```
$`197`
```

```
Linear mixed-effects model fit by maximum likelihood
```

```
Data: cipres
```

```
Log-likelihood: -17.8285
```

```
Fixed: Vol ~ Edad + Temp_inv + Temp_ver + 1
```

```
(Intercept) Edad Temp_inv Temp_ver
2.06656585 0.00689545 0.18154156 -0.15620499
```

```
Random effects:
```

```
Formula: ~1 | Region
```

```

      (Intercept)
StdDev: 1.264686e-05

  Formula: ~1 | Parcela %in% Region
      (Intercept)      Residual
StdDev:  0.2355302  0.0001154422

Variance function:
Structure: Power of variance covariate
  Formula: ~Edad
Parameter estimates:
  power
1.69193
Number of Observations: 78
Number of Groups:
      Region Parcela %in% Region
      15          37

```

### # 7.d- Identificar el mejor nulo

# El modelo nulo es el primer modelo que genera la función "dregde". Por lo tanto, se denomina "1"

```

RankAIC[c("1"),]
Global model call: lme.formula(fixed = Vol ~ Orient_N + Pendiente +
Altitud + Temp_ver + Temp_inv + Precipitacion + Densidad_cipres + Ed
ad, data = cipres, random = ~1 | Region/Parcela, weights = varPower(
form = ~Edad), method = "ML")
---
Model selection table
  (Int) df  logLik  AIC  delta weight
1 0.8499  5 -32.948  75.9  24.24      1
Random terms (all models):
'1 | Region', '1 | Parcela %in% Region'

```

# Es importante, informar el delta AIC del modelo nulo respecto del mejor modelo.

# En ese caso, el delta AIC es de 24.24

### # ¡¡ATENCIÓN!!

# No es lo mismo que,

```
RankAIC[1,]
```

# Esta sentencia nos muestra el modelo que quedó en primer lugar en el ranking. Es decir, el modelo modelo.

# En este caso, el modelo 197.

### # 8- Modelo estandarizado

# El mejor modelo incluye a los predictores Edad, Temp\_inv y Temp\_ver

# Sin embargo, no podemos comprar la magnitud de los efectos de los predictores.

# Para poder hacerlo debemos correr el modelo con los predictores estandarizados

#### # 8.a- Estandarizar predictores

# Cuando los predictores están medidos en escalas

# muy diferentes, como en este caso, una alternativa

# para interpretar el tamaño de los efectos es estandarizar

# en base a un desvío estándar (1 SD) haciéndolos comparables.



# Sin embargo, cuando trabajemos con variables binarias podría  
# estandarizarse por medio desvío estándar (Gelman, 2008; Grueber et al. 2011).

```
# Genero una matriz con las variables para poder estandarizar
x<-as.matrix(cipres[,c("Latitud", "Longitud", "Altitud", "Orient_N", "Orient_E",
  "Pendiente", "Precipitacion", "Temp_ver", "Temp_inv",
  "Densidad_cipres", "Edad")])
```

```
# z-score
x.cip<-scale(x, center = TRUE, scale = TRUE)
# La función "scale" es una función genérica que centra
# y/o escala las columnas de una matriz numérica.
# Los valores son centrados restándoles el valor medio de la columna (ignorando
# NAs)
# Los valores son escalados al dividirlos por su desvío estándar.
summary(x.cip)
```

```
cipres_standar<-cbind(cipres[,c("Vol", "Region", "Parcela")], x.cip)
summary(cipres_standar)
```

### # 8.b- Modelo estandarizado

```
# Escribimos del mejor modelo pero con los datos estandarizados (data=
cipres_standar)
modstand<-lme(Vol ~ Edad + Temp_inv + Temp_ver,
  data = cipres_standar,
  random = ~1 | Region/Parcela, weights = varPower(form = ~Edad),
  method = "ML")
```

```
modstand
fixef(modstand)
(Intercept)      Edad      Temp_inv      Temp_ver
  0.7965927    0.4394622    0.2244542   -0.2243091
```

# **IMPORTANTE:** la estandarización permite comparar la magnitud  
# de los efectos relativos entre los distintos predictores.

# Dado que los predictores están estandarizados,  
# podemos comparar la magnitud de los efectos de los predictores.

# **Tarea:** repetir el procedimiento multimodelo a partir de un modelo con  
# predictores estandarizados.  
# Observar los predictores que incluyen los mejores modelos y la magnitud de lo  
# efectos.

### # 9- Verificar el mejor modelo

# Volvemos a trabajar con la tabla de datos No estandarizados  
# Para ello, escribimos el modelo con la función lme y el método de REML

```
modFinal<-lme(Vol ~ Edad + Temp_inv + Temp_ver,
  data = cipres,
```

```
random = ~1 | Region/Parcela, weights = varPower(form = ~Edad),
method = "REML")
```

### # 9.a- Ajustados vs. los residuales del modelo

```
# Comparamos los modelos sin y con funciones de las varianzas
layout(matrix(1:3, 1, 3))
```

```
plot(x=FGloblme, y=EGloblme, xlab="Ajustados", ylab="Residuales
estandarizados", main="Modelo inicial,")
abline(a=0,b=0, col="violet", lw=2)
```

```
plot(x=FGlob, y=EGlob, xlab="Ajustados", ylab="Residuales estandarizados",
main="Modelo con varFunc")
abline(a=0,b=0, col="violet", lw=2)
```

```
EBest<-residuals(modFinal, type="normalized")
FBest<-fitted(modFinal)
plot(x=FBest, y=EBest, xlab="Ajustados", ylab="Residuales estandarizados",
main="Mejor modelo")
abline(a=0,b=0, col="violet", lw=2)
```

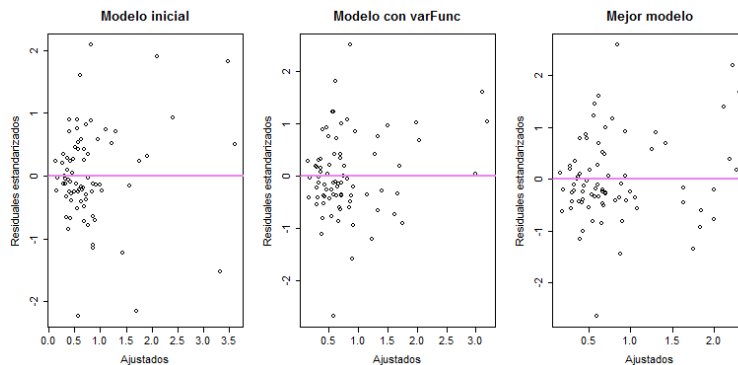


Figura 11 7: residuales estandarizados en relación a los valores ajustados por los modelos "GlobAlme" (inicial. Izq.), " modelVp6" (mejor modelo con función de la varianza. Centro) y, "modFinal" (mejor modelo según inferencia multimodelo. Der.).

### # 9.b- Residuales en función de las variables

```
layout (matrix(1:4, 2,2, byrow = TRUE))
plot(EBest~cipres$Region, main="Region", xlab="Region", ylab="Residuos
normalizados")
plot(EBest~cipres$Edad, main="Edad", xlab="Edad", ylab="")
abline(a=0,b=0, col="violet", lw=2)
plot(EBest~cipres$Temp_ver, main="Temp_ver", xlab="Temp_ver",
ylab="Residuos normalizados")
abline(a=0,b=0, col="violet", lw=2)
plot(EBest~cipres$Temp_inv, main="Temp_inv", xlab="Temp_inv", ylab="")
abline(a=0,b=0, col="violet", lw=2)
```

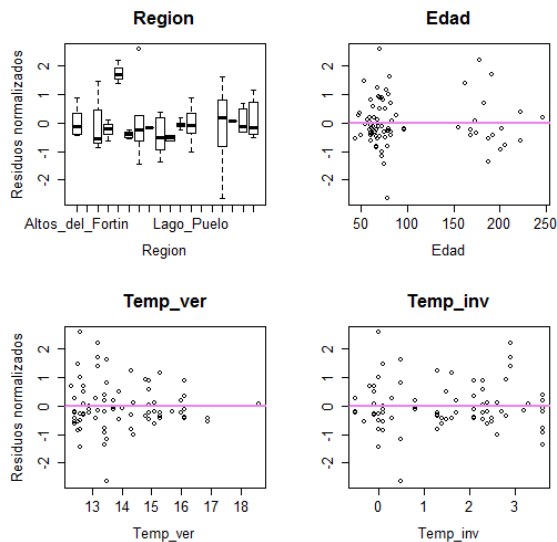


Figura 11 8: residuales del modelo "modFinal" en relación a las variables predictoras: Región, edad promedio de los árboles de ciprés, temperatura media de verano y temperatura media de invierno.

# 9.c- Normalidad

```
qqnorm(EBest, ylab="Errores estandarizados")
qqline(EBest)
plot(density(EBest))
rug(EBest)
```

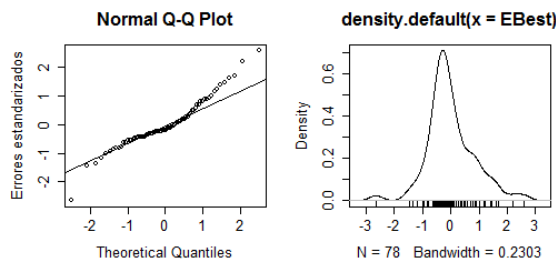


Figura 11 9: Gráfico de normalidad (cuantiles teóricos) para el mismo modelo "modFinal"

# 9.d- Distribución espacial de los residuos

```
mydata<- data.frame(EBest, cipres$Longitud, cipres$Latitud)
colnames(mydata) <- c("EBest", "Longitud", "Latitud")
coordinates(mydata) <- ~Longitud+Latitud
bubble(mydata, "EBest", col = c("black","red"),
       main = "Residuales", xlab = "Longitud",
       ylab = "Latitud")
```

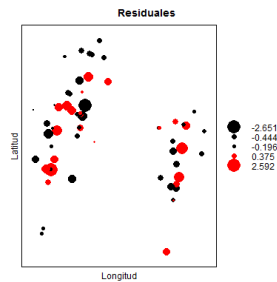


Figura 11 10: distribución espacial de los residuales normalizados del modelo "modFinal". Los puntos rojos indican valores positivos y los negro indican valores negativos. El tamaño de los puntos indica la magnitud del residual.

# Comparar la magnitud y distribución de los residuales respecto del modelo inicial.

### # 9.e- Variogramas

# Variograma inicial  
plot(Vario)

# Variograma final  
Variofinal <- variogram(EBest ~ 1, coord)  
plot(Variofinal)

# Discutir: ¿el modelo es adecuado?

### # 10- Presentamos el modelo

#### # 10.a- Salida del modelo

##### # Tarea

# A partir de  
summary(modFinal)

# Identificar:

- # - el parámetro que indica la bondad de ajuste.
- # - la variabilidad no explicada por el modelo entre Regiones
- # - la variabilidad no explicada por el modelo entre Parcelas
- # - la variabilidad residual del modelo
- # - las funciones de la varianza incluidas en el modelo y, sus parámetros

#### # 10.b- Factores aleatorios

# El modelo nos muestra los valores estimados para cada uno de los niveles de los factores aleatorios

ranef(modFinal)

# El modelo estimó una intercepción aleatoria para cada región y, para cada Parcela dentro de cada Región

# Podemos pedirle y graficar solo los valores estimados a nivel de Región

ranef(modFinal)\$Region  
plot(ranef(modFinal,level=1))

**# Tarea:**

# ¿Qué podemos decir de la variabilidad entre árboles de una misma Región y  
 # de la variabilidad entre árboles de distintas Regiones a partir de la magnitud  
 de los efectos aleatorios a nivel de Region?  
 # Este análisis, debería ser coherente con los desvíos estandar para los factores  
 aleatorios (a nivel Region) que observamos en el summary(modelo) en el puento  
 anterior.

**# 10.b- Predichos**

# Genero un objeto donde copio la tabla de datos  
 cipres\_pred<-cipres

# En esta nueva tabla agrego dos columnas: "pred\_Region" con los predichos a  
 nivel Region (level=0) y,  
 # "pred\_Arbol" con los predichos a nivel de árbol.  
 cipres\_pred\$pred\_Arbol<-predict(modFinal, newdata=cipres\_pred)

**# 10.d- Grafico de ajuste**

```
par(mfrow=c(1,1))
plot(cipres_pred$pred_Arbol, cipres_pred$Vol, xlim=c(0,4), ylim=c(0,4), main="A
nivel de árbol", xlab="Vol Predicho", ylab="Vol observados")
abline(a=0,b=1, col="violet", lw=2)
```

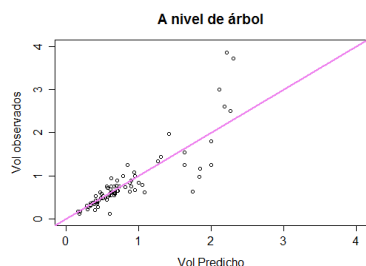


Figura 11 11: valores observados de volumen de ciprés (m<sup>3</sup>/árbol) en función de los valores ajustados por el modelo "modFinal" según el método de inferencia multimodelo.

**# 11- Graficar el modelo final**

# Utilizaremos los valores NO estandarizados

**# 11.a- Gráficos 2D**

```
attach(cipres_pred)
# La función "attach" adjunta la tabla de datos de modo
# que ya no es necesario indexarla usando por ejemplo cipre$.
# La función opuesta es "dettach()"
par(mfrow=c(2,2))
par(mar=c(4, 5, 1, 1))
par(oma=c(0, 0, 2, 0))
```

# La función "par" indica parámetros del área gráfica  
 # La cantidad de cuadrantes ("mfrow" es similar a la función layout(matrix(...))  
 que usamos antes).

# Los márgenes ("mar" indicando el tamaño del margen inferior, izquierdo, superior y derecho).

# La función "par" admite numerosos parámetros: consultar ?par

### ## Volumen en función de la Edad

# Graficamos los valores observados

```
plot(Edad, Vol,
     pch=16, type="p", col = "gray",
     ylab="", xlab="", cex.lab=0.9, cex.axis=0.8, #mar=margins,
     xlim=c(min(Edad), max(Edad)),
     ylim=c(min(Vol), max(Vol)))
```

```
title(xlab = "Edad (años)",
      ylab = expression(paste("Vol (", m^3, ")", sep = ""))),
points(Edad, pred_Arbol, pch=16, type="p", col = "black")
legend("topleft", c("Observ", "Pred"), pch=16, col=c("gray", "black"))
fixef(modFinal)
```

### ## Volumen en función de la Temperatura de invierno

```
plot(Temp_inv, Vol,
     pch=16, type="p", col = "gray",
     ylab="", xlab="",
     cex.lab=0.9, cex.axis=0.8,
     xlim=c(min(Temp_inv), max(Temp_inv)),
     ylim=c(min(Vol), max(Vol)))
```

```
title(xlab = expression(paste("Temp invierno (°C)", sep = "")),
      ylab = expression(paste("Vol (", m^3, ")", sep = ""))),
points(Temp_inv, pred_Arbol, pch=16, type="p", col = "black")
```

### ## Volumen en función de la Temperatura de veranos

```
plot(Temp_ver, Vol,
     pch=16, type="p", col = "gray",
     ylab="", xlab="",
     cex.lab=0.9, cex.axis=0.8,
     xlim=c(min(Temp_ver), max(Temp_ver)),
     ylim=c(min(Vol), max(Vol)))
```

```
title(xlab = expression(paste("Temp verano (°C)", sep = "")),
      ylab = expression(paste("Vol (", m^3, ")", sep = ""))),
points(Temp_ver, pred_Arbol, pch=16, type="p", col = "black")
title(main="Valores Retransformados", outer=TRUE)
```

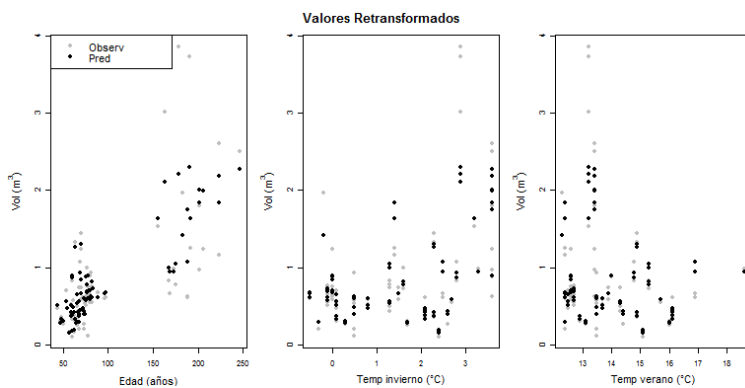


Figura 11 12: volumen de ciprés (m<sup>3</sup>) en función de la Edad promedio (Izq.) de la temperatura media de verano (Centro) y la temperatura media de invierno (Der.). Los puntos grises muestran los valores observados y los negros, los predichos por el modelo "modFinal".

### # 11.b- Gráficos 3D

```

par(mfrow=c(1,1))
library(scatterplot3d)
s3d<-scatterplot3d(cipres$Temp_inv,
  cipres$Edad,
  cipres$Vol,
  pch=16, type="h", grid=TRUE, box = FALSE,
  color = "gray", mar=c(3,3,0,2), #highlight.3d=TRUE,
mar=c(5,3,4,3)+0.1, default
  main="",
  ylab="Edad (años)", xlab="Temp inv (°C)", zlab=expression(paste("Vol
(", m^3, ")", sep = ""))), cex.lab=1, cex.axis=0.8,
  ylim=c(min(cipres$Edad), max(cipres$Edad)),
  xlim=c(min(cipres$Temp_inv), max(cipres$Temp_inv)),
  zlim=c(min(cipres$Vol), max(cipres$Vol)))

# Los que predice el modelo general
s3d$points3d(cipres$Temp_inv,
  cipres$Edad,
  cipres_pred$pred_Arbol,
  pch=16, col="green", type="p")

```

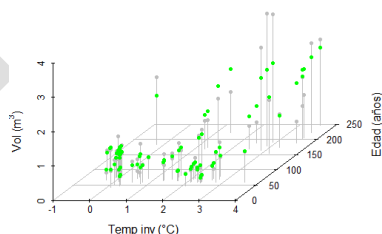


Figura 11 13: volumen de ciprés (m<sup>3</sup>) en función de la temperatura media de invierno (°C) y la edad promedio (años). Los puntos grises muestran los valores observados y los verdes, los predichos por el modelo "modFinal".

### # 12- Resumen

- # En este caso ajustamos un modelo predictivo de la productividad del ciprés (vol de madera).
- # Generamos un modelo global (incluyendo todas los predictores según nuestra hipótesis de trabajo).
- # Comparamos modelos con igual componente fija y distintos factores aleatorios incluyendo factores aleatorios cruzados y pendientes aleatorias.
- # Incorporamos funciones de la varianza para corregir problemas de Homocedasticidad (heterogeneidad en las varianzas).
- # En este caso, los supuestos del modelo se cumplen.

- # Utilizamos como marco inferencial la "inferencia multimodelo"
- # A partir del modelo global de mejor ajuste generamos todos los modelos posibles de combinar los predictores fijos.
- # Estandarizamos los daos para comparar la magnitud de los efectos relativos entre los distintos predictores.
- # Esto es espacialmente útil cuando las magnitudes en que son medidas las variables son muy distintas.

- # Continuamos el análisis escribiendo el mejor modelo.
- # Verificamos la validez del mejor modelo
- # Analizamos los parámetros aleatorio y fijos estimados por el modelo
- # Generamos gráficos que nos muestran los valores observados y los estimados por el modelo.

### # 13- Agradecimiento

- # Datos gentilmente provistos por Juan Pablo Langlois y Lucas A. Garibaldi de la Facultad de Agronomía de la Universidad de Buenos Aires. Langlois J. P.. 2013. Silvicultura de ciprés de la cordillera: condiciones de sitio y rendimiento. Tesis para obtener el grado de Ingeniero agrónomo-FAUBA. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.



# Modelos lineales generalizados mixtos [GLMM]

## Contenidos

<b># CAPÍTULO 12: Modelos lineales generalizados mixtos - Binomial</b> .....	<b>194</b>
<b># 1- Configuración inicial</b> .....	<b>194</b>
<b># 2- El caso: Características reproductivas de <i>Baccharis dracunculifolia</i></b> .....	<b>194</b>
# 2.a- Pregunta de interés .....	194
# 2.b- Variable respuesta.....	194
# 2.c- Variable/s predictoras de interés.....	194
# 2.d- Identificar jerarquías.....	194
<b># 3- Leer el archivo de datos</b> .....	<b>195</b>
<b># 4- Definir la variable respuesta binomial (0-1)</b> .....	<b>195</b>
<b># 5- Exploramos los datos</b> .....	<b>195</b>
<b># 6- Planteamos el modelo</b> .....	<b>196</b>
# 6.a- Modelo glm.....	197
# 6.b- Dispersión binomial glm .....	197
# 6.c- Modelos glmer .....	198
<b># 7- Estimación del modelo final</b> .....	<b>201</b>
# 7.a- Parámetros estimados por el modelo.....	201
# 7.b- Intervalos de confianza.....	202
# 7.c- Variación de los efectos aleatorios.....	202
# 7.d- Graficar los efectos aleatorios.....	203
# 7.e- Estimar el error estándar de los efectos fijos y aleatorios.....	203
<b># 8- Representación gráfica de este modelo general</b> .....	<b>204</b>
<b># 9- Modelos glmmADMB</b> .....	<b>206</b>
<b># 10- Resumen</b> .....	<b>207</b>
<b># 11- Tarea</b> .....	<b>208</b>
<b># 12- Agradecimiento</b> .....	<b>208</b>

## # CAPÍTULO 12: Modelos lineales generalizados mixtos - Binomial

### # 1- Configuración inicial

Definir el directorio de trabajo

# setwd("D:\\Mis documentos")

### # 2- El caso: Características reproductivas de *Baccharis dracunculifolia*

## Pque. Nacional “El Palmar” (Entre Ríos)

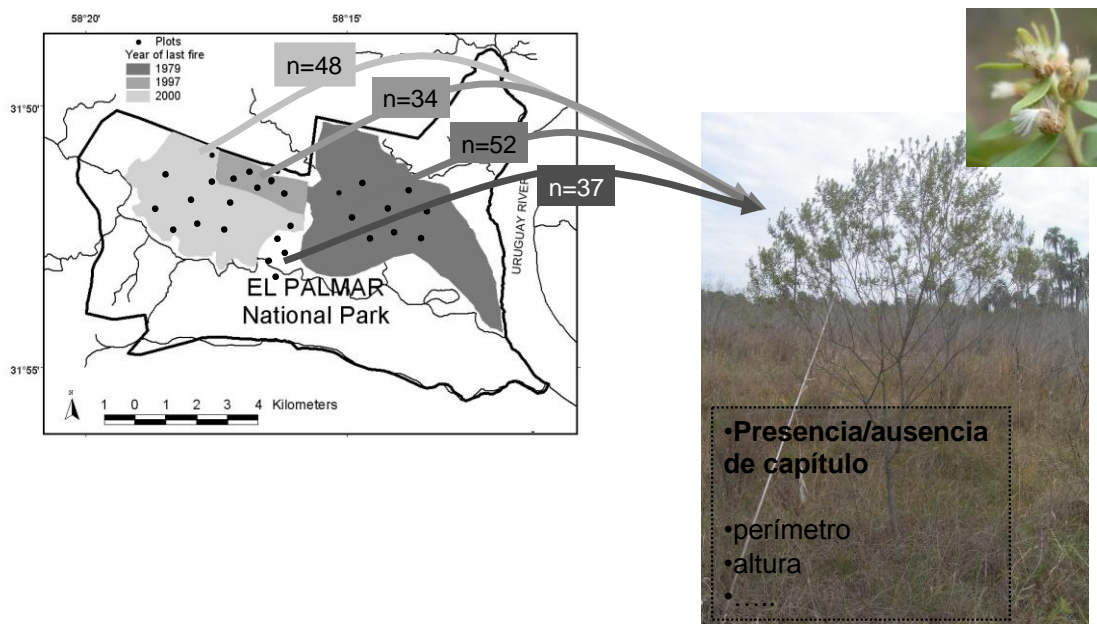


Figura 12. 1: diseño del experimento

### # 2.a- Pregunta de interés

# ¿La probabilidad de que un arbusto de *Baccharis* florezca, depende de la altura?

### # 2.b- Variable respuesta

# presencia/ausencia de Capítulos (N=1)

### # 2.c- Variable/s predictoras de interés

# Altura de las plantas (Fig. 12.1)

### # 2.d- Identificar jerarquías

# ¿Cuáles son las escalas y qué variables se midieron en cada uno?

# Sitios (4): 1965, 1979, 1997, 2000 (año desde el último incendio)

# Las plantas dentro de un sitio no son independientes

# Las variables fueron medidas a nivel de planta

### # 3- Leer el archivo de datos

```
# En este caso, la tabla de datos "12_Baccharis.csv"
baccharis <- read.csv("12_Baccharis.csv")
summary(baccharis)
baccharis$fSitio <- factor(baccharis$Sitio)
# Generamos una nueva variable que contiene a "Sitio" como factor: "fSitio"
summary(baccharis)
```

```
Sitio          ind          altura          cap          N
Min.:1965      d3-160:2      Min.:40.0      Min.:0.0      Min.:0.00
1st Qu.:1979  d3-200:2      1st Qu.:120.0  1st Qu.:0.0   1st Qu.: 0.00
Median :1997  d4-200:2      Median :160.0  Median :178.0 Median :12.00
Mean :1986   d4+200:2      Mean :164.1    Mean :633.8   Mean :10.23
3rd Qu.:2000 d5-120:2      3rd Qu.:200.0  3rd Qu.:726.0 3rd Qu.:16.00
Max. :2000   d5-160:2      Max. :250.0    Max. :9450.0  Max. :20.00
                (Other):159
```

### # 4- Definir la variable respuesta binomial (0-1)

```
# Los valores posibles para la variable respuesta son: 1 (presencia), 0 (ausencia)
de capítulos (cap)
# En la columna cap, a todos los valores de cap>0 les asigno 1
# Generamos dos nuevas columnas, una con valores = 1 para los baccharis con
capítulos: sicap
# y otra con valores = 1 para los baccharis sin capítulos: nocap
baccharis$sicap <- ifelse(baccharis$cap>0, 1, 0)
baccharis$nocap <- ifelse(baccharis$sicap==0, 1, 0)
```

```
summary(baccharis)
Sitio          ind          altura          cap          N          sicap          nocap
Min.:1965      d3-160:2      Min.:40.0      Min.:0.0      Min.:0.0      Min.:0.0      Min.:0.0
1stQu.:1979   d3-200:2      1stQu.:120.0   1stQu.:0.0     1stQu.: 0.00   1stQu.:0.0     1stQu.:0.0
Median:1997   d4-200:2      Median:160.0   Median:178.0   Median :12.00   Median:1.0     Median:0.0
Mean:1986    d4+200:2      Mean:164.1     Mean:633.8     Mean:10.23     Mean:0.7135    Mean:0.2
3rd Qu.:2000 d5-120:2      3rdQu.:200.0   3rdQu.:726.0   3rdQu.:16.00   3rdQu.:1.0000  3rdQu.:1.0
Max.:2000    d5-160:2      Max.:250.0     Max.:9450.0    Max.:20.00     Max.:1.0000    Max.:1.0
                (Other):159
```

```
View(baccharis)
```

```
# Unimos las dos variable nuevas: sicap y nocap en una tabla.
cap.bi <- cbind(baccharis$sicap, baccharis$nocap) # la función "cbind" une (bind)
columnas (c)
# ATENCIÓN: ¡Las dos columnas conjuntamente definen la variable respuesta!
```

### # 5- Exploramos los datos

#### # Distribución de la variable respuesta

```
# Generamos un vector que contiene la cantidad de niveles que tiene la variable
sitio.
```

```
cl <- length(levels(baccharis$fSitio))
```

```
# Lo usaremos para definir la cantidad de colores en el gráfico y en la leyenda.
```

```
# La cantidad de capítulos en función de la altura del arbusto de Baccharis
with(baccharis, plot(altura, cap, col=1:cl, pch=16, main="Altura", ylab="Presencia
de capítulos"))
```

```
legend("topleft", levels(baccharis$fSitio), col=1:cl, pch=16, bty="o", bg="light
grey")
```

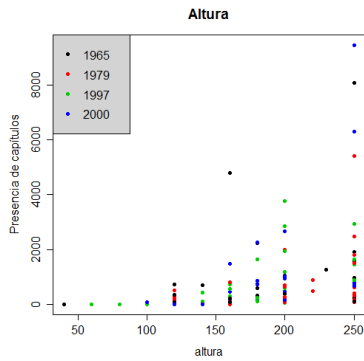


Figura 12. 2: cantidad de capítulos en relación a la altura de las plantas de *Baccharis dracunculifolia*. Los colores indican distintos sitios de muestreo identificados con el año de ocurrencia del último incendio: 1965 (negro), 19779 (rojo), 1997 (verde), 2000 (azul).

# Ahora, graficamos si tiene o no capítulos en función de la altura del arbusto de Baccharis

```
with(baccharis, plot(altura,sicap, col=1:cl, pch=16, main="Altura",
ylab="Presencia de capítulos"))
```

```
legend("bottomright",levels(baccharis$fSitio), col=1:cl, pch=16,
bty="o",bg="light grey" )
```

```
?legend
```

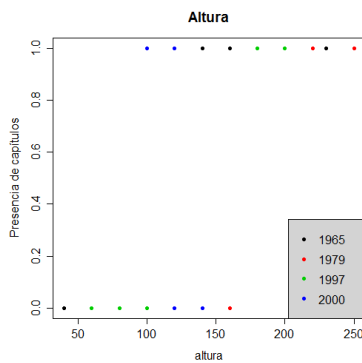


Figura 12. 3: probabilidad de presencia (presencia/ausencia) de capítulos en relación a la altura de las plantas de *Baccharis dracunculifolia*. Los colores indican distintos sitios de muestreo identificados con el año de ocurrencia del último incendio: 1965 (negro), 19779 (rojo), 1997 (verde), 2000 (azul).

## # 6- Planteamos el modelo

```
# El objetivo es predecir la presencia de capítulos en el arbusto i (N=1)
```

```
# teniendo en cuenta el efecto de la altura y del perímetro y,
```

```
# considerando el sitio donde se encuentra cada Baccharis
```

```
# En este caso, la variable respuesta puede tomar valores 0 ó 1,
```

```
# suponemos que los residuos con N=1 no tendrán una distribución
```

```
# normal sino, una distribución binomial.
```

```
# Las funciones glm (para modelos de factores fijos) y
```

```
# lmer (para modelos mixtos) admiten el argumento "family"
```

```
# Este argumento nos permite indicar la distribución
```

```
# (binomial, poisson, gamma, gaussian) de los residuales.
```

```
# Recordamos que la variable respuesta contiene dos columnas:
```

# "éxitos" (i.e arbustos con capítulos [,1] ) y "fracasos" (i.e. arbustos sin capítulos [,2]).

```
head(cap.bi)
      [,1] [,2]
[1,]    1    0
[2,]    0    1
[3,]    1    0
[4,]    0    1
[5,]    1    0
[6,]    1    0
```

# "cap.bi" será interpretado como la proporción de arbustos con capítulos.

### # 6.a- Modelo glm

# Recordamos un modelo glm. Es decir, sin efectos aleatorios.

# ATENCIÓN: este modelo asume erróneamente, que todas las observaciones son independientes.

```
M<-glm(cap.bi ~ altura, family=binomial(link="logit"), data = baccharis)
```

### # 6.b- Dispersión binomial glm

# En el caso de la distribución binomial,

# "phi" es el parámetro que estima la dispersión

# El estadístico "X2" (see lee "Chi cuadrado") de Pearson

# es la sumatoria de los residuales

# de Pearson elevados al cuadrado:

```
X2 <- sum(residuals(M, type = "pearson")^2)
```

# Si dividimos este valor por los grados de libertad de los residuales

# del modelo estimamos el parámetro phi dispersión

```
phi <- X2/M$df.residual
```

```
phi
```

```
[1] 0.5438443
```

# El phi estimado también podemos obtenerlo a partir

# de la "deviance residual" con un resultado similar.

```
phi <- M$deviance/M$df.residual
```

```
phi
```

```
[1] 0.4973325
```

# Al final del summary(modelo) para la función glm se encuentra

# el "Residual deviance" y los grados de libertad

```
summary(M)
```

```
Null deviance: 204.869 on 170 degrees of freedom
Residual deviance: 84.049 on 169 degrees of freedom
AIC: 88.049
```

```
Number of Fisher Scoring iterations: 7
```

# En este caso "Residual deviance" es mucho menor

# a grados de libertad (84.049 << 169).

# Por lo tanto, el parámetro de dispersión phi será << a 1

```
84.049 / 169
```

```
0.4973314
```

# **ATENCIÓN:** en este caso particular con una distribución binomial  
 # con  $N=1$ , NUNCA hay sobre-dispersión  
 # Prueben el script cuando planteen un modelo con datos con  $N$  distinto de 1

### # 6.c- Modelos glmer

# La función glmer del paquete lme4 permite ajustar modelos jerárquicos  
 # con distintas distribuciones.  
 library("lme4")  
 # Recordamos que las variables "altura" y "perímetro" tienen una correlación de 0.75.

#### # 6.c.1- Modelo glmer inicial

```
M.0<-glmer(cap.bi ~ altura +(1|fSitio),
            family=binomial(link="logit"),
            data = baccharis)
```

# el "warning message" que aparece aquí aparentemente es un  
 # "falso positivo de error de convergencia" aún no resuelto  
 # por los programadores del paquete en esta versión de lme4  
 # pueden revisar esta serie de mails de discusión sobre el tema  
 # si están interesados  
 # <https://stat.ethz.ch/pipermail/r-sig-mixed-models/2014q2/021967.html>  
 # podemos seguir con la práctica sin problemas

```
# M.0<-glmer(cap.bi ~ altura +(1|fSitio),
#             family=binomial(link="logit"),
#             control=glmerControl(check.conv.grad=
# .makeCC("warning", tol = 0.1, relTol = NULL)),
#             data = baccharis)
```

#### # 6.c.2- Dispersión binomial glmer

# Evaluamos la sobre-dispersión  
 # Recordamos: el supuesto de esta distribución es que la varianza estimada es menor a la media estimada.  
 # Para estos modelos, evaluamos sobre-dispersión de acuerdo con la función propuesta por Bolker et al. (2009)  
 overdisp\_fun<- function(model) {  
 ## number of variance parameters in  
 ## an n-by-n variance-covariance matrix  
 vpars<- function(m) {  
 nrow(m)\*(nrow(m)+1)/2  
 }  
 model.df<- sum(sapply(VarCorr(model),vpars))+length(fixef(model))  
 (rdf<- nrow(model@frame)-model.df)  
 rp<- residuals(model)  
 Pearson.chisq<- sum(rp^2)  
 prat<- Pearson.chisq/rdf  
 pval<- pchisq(Pearson.chisq, df=rdf, lower.tail=FALSE,log.p=TRUE)

```

c(chisq=Pearson.chisq,ratio=prat,p=exp(pval))
}
# Una vez que le asignamos la función al objeto,
# queda en la memoria temporal de R y la utilizamos con el modelo

```

```

overdisp_fun(M.0)
  chisq      ratio      p
72.8035241  0.4333543  1.0000000

```

```

# Valores menores a 1 en el ratio indican que no hay sobre-dispersión.
# Recordamos que no hay sobre-dispersión para el caso especial cuando N=1,
idem arriba.

```

### # 6.c.3- Estimación del modelo

```
summary(M.0)
```

```

Generalized linear mixed model fit by maximum likelihood (Laplace A
approximation) [glmerMod]

```

```

Family: binomial ( logit )
Formula: cap.bi ~ altura + (1 | fSitio)
Data: baccharis

```

```

AIC      BIC    logLik deviance df.resid
86.6     96.0   -40.3    80.6     168

```

```

Scaled residuals:
  Min       1Q   Median       3Q      Max
-2.60123 -0.11421  0.04026  0.19965  1.90643

```

```

Random effects:
  Groups Name      Variance Std.Dev.
fSitio (Intercept) 0.8508  0.9224
Number of obs: 171, groups: fSitio, 4

```

```

# El modelo asume que los residuos tienen distribución binomial.
# Pero, para el factor aleatorio sitio asume una distribución normal
# con media cero y varianza = 0.9224^2 = 0.8508
# Observamos que en el caso de la distribución binomial no
# hay una varianza residual del modelo. La varianza = n*p*(1-p).

```

```

Fixed effects:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  -9.55892    1.95056  -4.901 9.55e-07 ***
altura       0.07521    0.01401   5.370 7.88e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Correlation of Fixed Effects:
      (Intr)
altura -0.959

```

### # 6.c.4- Calcular la correlación entre estimadores de los efectos fijos #####

```

# ¿Cómo se calcula la correlación entre estimadores de los efectos fijos?
# a partir de la matriz de varianzas y covarianzas:

```

```

vcov(M.0)
2 x 2 Matrix of class "dpoMatrix"
      (Intercept)      altura

```

```
(Intercept) 3.80467366 -0.026197686
altura      -0.02619769  0.000196193
```

```
-0.026197686/prod(sqrt(diag(vcov(M.0))))
[1] -0.958876
```

# La correlación es entre los valores estimados de los efectos fijos.  
# No entre variables, de hecho aquí hay una sola variable predictora.

#### # 6.c.5- Obtener el R<sup>2</sup> del modelo #####

# Nakagawa et al. (2013) proponen una manera de obtener el R<sup>2</sup> de un modelo de efectos mixtos.

# El trabajo muestra ejemplos para los casos de distribución normal, binomial y poisson ajustados con la función glmer (lme4).

```
# Extraemos los valores ajustados por nuestro modelo
# fixef() extrae los coeficientes de los efectos fijos.
# mF@pp$X devuelve la matriz de diseño de los efectos fijos
Fixed <- fixef(M.0)[2] * M.0@pp$X[, 2]
```

# En este caso, el efecto fijo es solo "Altura". Si el modelo tuviera más predictores debemos hacer la sumatoria del producto de los predictores

```
# Ejemplo, Fixed <- fixef(Modelo)[2] * Modelo@pp$X[, 2] + fixef(Modelo)[3] *
Modelo@pp$X[, 3] + fixef(Modelo)[4] * Modelo@pp$X[, 4]
```

```
# Calculamos la varianza de los valores ajustados
VarF <- var(Fixed)
```

```
# Una forma alternativa para obtener el mismo resultado
VarF <- var(as.vector(fixef(M.0) %*% t(M.0@pp$X)))
```

# R<sup>2</sup> marginal

# El R<sup>2</sup> marginal describe la proporción de varianza explicada por el factor/es fijo/s.

```
VarF/(VarF + VarCorr(M.0)$fSitio[1] + pi^2/3)
[1] 0.8227914
```

# R<sup>2</sup> condicional

# El R<sup>2</sup> condicional describe la proporción de la varianza explicada tanto por los factores fijos como aleatorios.

```
(VarF + VarCorr(M.0)$fSitio[1])/(VarF + VarCorr(M.0)$fSitio[1] + pi^2/3)
[1] 0.8592031
```

#### # 6.c.6- Modelo sin ordenada al origen

# Por si queremos aprender a estimar un modelo sin ordenada (Es decir, ordenada al origen igual a cero)

```
M.0b<-glmer(cap.bi ~ altura-1 +(1|fSitio),family=binomial(link="logit"), data =
baccharis)
```



```
anova(M.0, M.0b)
Data: baccharis
Models:
M.0b: cap.bi ~ altura - 1 + (1 | fSitio)
M.0: cap.bi ~ altura + (1 | fSitio)
      Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
M.0b  2  101.949 108.233 -48.975  97.949
M.0   3   86.561  95.986 -40.281  80.561 17.388      1 3.047e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Cuando comparamos el modelo sin ordenada, el ajuste del modelo empeora y la varianza aumenta.

### # 7- Estimación del modelo final

# Recordamos la salida anterior

```
summary(M.0)
```

# ¿Qué otra información contiene el modelo?

```
class(M.0)
```

```
[1] "glmerMod"
attr(,"package")
[1] "lme4"
```

# Observar que el modelo "M.0" es un objeto de R de clase "lmerMod".

# para explorar su estructura usamos la función "slotNames":

```
names(attributes(M.0))
```

```
# ó, slotNames(M.0)
```

```
[1] "resp"      "Gp"        "call"      "frame"     "flist"     "cnms"      "lower"     "
theta"      "beta"      "u"         "devcomp"   "pp"        "optinfo"
```

# Para ver el contenido de los argumentos anteriores usaremos "@" (M.0@...)

# Por ejemplo,

```
M.0@call # observamos el modelo
```

```
glmer(formula = cap.bi ~ altura + (1 | fSitio), data = baccharis,
      family = binomial(link = "logit"))
```

```
M.0@beta
```

```
[1] -9.55892419  0.07521387
```

# Los modelos merMod tienen un número de métodos disponibles - demasiados para enumerar aquí.

# Vamos a utilizar solo algunos de los más comunes:

```
methods(class = "merMod")
```

### # 7.a- Parámetros estimados por el modelo

```
coef(M.0)
```

```
$fSitio
      (Intercept)  altura
1965 -10.316133  0.07521387
1979  -8.992573  0.07521387
1997 -10.356480  0.07521387
2000  -8.617975  0.07521387
```

# Con distintas ordenadas para cada sitio.

# Veamos solo los efectos fijos del modelo general

```
fixef(M.0)
```

```
(Intercept)      altura
-9.55892419  0.07521387
```

# O, solo los aleatorios

```
ranef(M.0)$fSitio
```

```
(Intercept)
1965 -0.7572091
```

```
1979 0.5663512
1997 -0.7975558
2000 0.9409497
```

# la función "ranef" muestra la diferencia de las intercepciones  
# aleatorias respecto de la intercepción general del modelo.

### # 7.b- Intervalos de confianza

# La función "confint" permite observar el intervalos de confianza de los  
parámetros del modelo

```
confint(M.0, level = 0.95)
      2.5 %      97.5 %
.sig01 0.000000      Inf
(Intercept) -Inf -6.3383292
altura 0.0518324 0.1077172
```

# Podemos observar que:

# - .sig01, estima la variabilidad en el efecto aleatorio que, en este caso es muy grande.

# Esto indica que podemos tener falta de precisión entre nuestros grupos (sitios en este caso) - ya sea debido a que el efecto grupo es pequeño entre los grupos, tenemos muy pocos grupos para obtener una estimación más precisa, tenemos muy pocas unidades dentro de cada grupo, o una combinación de todos lo anterior.

# - nuestro parámetro de efecto fijo (Altura) no se superpone a 0. Esto indica que hay evidencia de un efecto.

### # 7.c- Variación de los efectos aleatorios

# Cuando escribimos un modelo de efectos mixtos estamos interesados en la  
variación a nivel del grupo (en este caso, los sitios) en el modelo.

# Sin embargo, no es tan claro cómo explorar esta variación a nivel de grupo a partir de los resultados de la función "summary(modelo)".

# Lo que obtenemos de esta salida es la varianza y el desvío estándar del efecto del grupo, pero no conseguimos efectos para los grupos individuales.

# Aquí es donde la función ranef viene muy bien.

```
ranef(M.0)
```

# La función "ranef" nos da las intersecciones de cada sitio, pero no mucha más información -

# por ejemplo, la precisión de estas estimaciones.

# Para hacer eso, necesitamos algunos comandos adicionales:

```
re1 <- ranef(M.0, condVar=TRUE) # generamos un objeto que contiene los
efectos aleatorios
```

```
class(re1)
[1] "ranef.mer"
```

# El objeto generado con la función "ranef" es una lista que contiene una tabla de datos para cada nivel de grupo.

# Esta tabla de datos contiene los efectos aleatorios para cada grupo (en este caso sólo tenemos una intercepción para cada sitio).

# Cuando le pedimos a lme4 la varianza condicional de los efectos aleatorios (condVar=TRUE) se almacena en un atributo de la tabla de datos como una lista de matrices de varianza-covarianza.

# La siguiente sentencia no muestra las varianzas para cada sitio:

```
attr(re1[[1]], which = "postVar")
```

```
[1,] 1 0.2625752
```

```
[1,] 2 0.2033229
```

```
[1,] 3 0.342756
```

```
[1,] 4 0.209284
```

# ó,

```
# attr(ranef(M.0, condVar = TRUE)[[1]], "postVar")
```

# que da el mismo resultado

#### # 7.d- Graficar los efectos aleatorios

```
library(lattice)
```

```
re1 <- ranef(M.0, condVar=TRUE, whichel = "fSitio")
```

```
print(re1)
```

```
dotplot(re1)
```

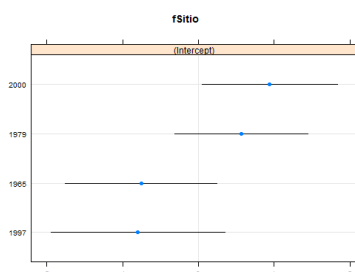


Figura 12. 4: efecto aleatorio (diferencia respecto de la intercepción) estimado por el modelo "M.0" para cada uno de los sitios (1965, 1979, 1997, 2000).

# Este gráfico muestra los efectos de cada sitio en la probabilidad de que baccharis haya florecido (la diferencia respecto del estimado general)

# así como sus errores estándar que ayudan a identificar cuán distintos son los sitios entre sí (línea representa el intervalo de predicción al 95 %).

# El modelo estima que la intercepción de los años 1965 y 1997 fue inferior que la intercepción general del modelo y, que la intercepción de los años 1979 y 2000 fue superior que la intercepción general del modelo.

# La interpretación de efectos aleatorios no es tan sencilla. Sin embargo, la lectura de algunos libros puede ayudar:

# Gelman and Hill 2006 - Data Analysis Using Regression and Multilevel/Hierarchical Techniques

# John Fox - An R Companion to Applied Regression web appendix

#### # 7.e- Estimar el error estándar de los efectos fijos y aleatorios

```
library(arm)
```

```
se.coef(M.0)
```

```
# $fixef
```

```
# [1] 1.95055727 0.01400689
```

```
#
```

```
# $fSitio
```

```
# (Intercept)
```

```
# 1965 0.5124210
# 1979 0.4509134
# 1997 0.5854537
# 2000 0.4574757
se.fixef (M.0)
se.ranef (M.0)
```

### # 8- Representación gráfica de este modelo general

```
plot(baccharis$altura, baccharis$sicap, xlab="altura", ylab="Prob. de capítulos en
Baccharis")
```

```
# Utilizamos los efectos fijos para obtener los predichos según los efectos fijos
(sin considerar distintas ordenadas por sitio)
pred<-fixef(M.0)[1]+fixef(M.0)[2]*baccharis$altura
# La familia binomial tienen por defecto la función de enlace logit.
# Entonces, "pred" son los valores predichos de "p"
# (probabilidad de llegar al estado reproductivo en un arbusto) en escala logit
# Debemos retrotransformar "pred" a la escala "original"
p.gral<-exp(pred)/(1+exp(pred))
```

```
# Alternativamente,
library(boot)
inv.logit(pred)
```

```
ord<-order(baccharis$altura) # generamos el vector con la posición de las
alturas ordenadas de menor a mayor
lines(baccharis$altura[ord], p.gral[ord], lwd=3, col=1)
```

```
p.up<-exp(pred+1.96*0.8508)/(1+exp(pred+1.96*0.8508))
lines(baccharis$altura[ord], p.up[ord], lwd=1,lty=2,col=1) #
p.low<-exp(pred-1.96*0.8508)/(1+exp(pred-1.96*0.8508))
lines(baccharis$altura[ord], p.low[ord], lwd=1,lty=2,col=1) #
# La línea representa la probabilidad de encontrar un arbusto de baccharis con
capítulos en cualquiera de los sitios.
# Sin embargo, dependiendo del sitio, la probabilidad puede variar entre las dos
líneas punteadas externas
```

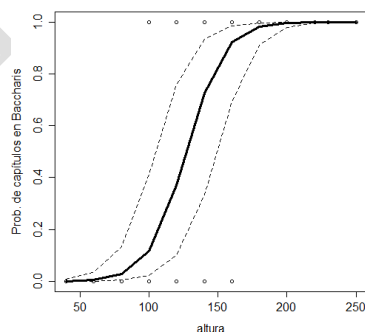


Figura 12. 5: probabilidad de presencia de capítulos en función de la altura de plantas de *Baccharis dracunculifolia* estimada por el modelo "M.0". La línea llena muestra la probabilidad de encontrar un arbusto

de baccharis con capítulos en cualquiera de los sitios estimado por el modelo y las líneas punteadas el intervalo de confianza de la estimación.

```
layout(matrix(1:4,2,2))
with(subset(baccharis, fSitio=="1965"),plot(baccharis$altura, baccharis$sicap,
xlab="altura",main="Sitio 1965", ylab="Prob. de capítulos en Baccharis"))
lines(baccharis$altura[ord], p.gral[ord], lwd=1, col=1)
```

```
p1965<-coef(M.0)[[1]][1,1]+fixef(M.0)[2]*baccharis$altura
p.5<-exp(p1965)/(1+exp(p1965)) # re transformamos el logit
ord<-order(baccharis$altura) # generamos el vector con la posición de las
alturas ordenadas de menor a mayor. Como hicimos anteriormente
lines(baccharis$altura[ord], p.5[ord], lwd=2, col=1)
```

```
with(subset(baccharis, fSitio=="1979"),plot(baccharis$altura, baccharis$sicap,
xlab="altura",main="Sitio 1979", ylab="Prob. de capítulos en Baccharis"))
lines(baccharis$altura[ord], p.gral[ord], lwd=1, col=1)
p1979<-coef(M.0)[[1]][2,1]+fixef(M.0)[2]*baccharis$altura
p.9<-exp(p1979)/(1+exp(p1979)) # re transformamos el logit
ord<-order(baccharis$altura) # generamos el vector con la posición de las
alturas ordenadas de menor a mayor. Como hicimos anteriormente
lines(baccharis$altura[ord], p.9[ord], lwd=3, col=2) #
```

```
with(subset(baccharis, fSitio=="1997"),plot(baccharis$altura, baccharis$sicap,
xlab="altura",main="Sitio 1997", ylab="Prob. de capítulos en Baccharis"))
lines(baccharis$altura[ord], p.gral[ord], lwd=1, col=1)
p1997<-coef(M.0)[[1]][3,1]+fixef(M.0)[2]*baccharis$altura
p.7<-exp(p1997)/(1+exp(p1997)) # re transformamos el logit
ord<-order(baccharis$altura) # generamos el vector con la posición de las
alturas ordenadas de menor a mayor. Como hicimos anteriormente
lines(baccharis$altura[ord], p.7[ord], lwd=3, col=3) #
```

```
with(subset(baccharis, fSitio=="2000"),plot(baccharis$altura, baccharis$sicap,
xlab="altura",main="Sitio 2000", ylab="Prob. de capítulos en Baccharis"))
lines(baccharis$altura[ord], p.gral[ord], lwd=1, col=1)
p2000<-coef(M.0)[[1]][4,1]+fixef(M.0)[2]*baccharis$altura
p.0<-exp(p2000)/(1+exp(p2000)) # re transformamos el logit
ord<-order(baccharis$altura) # generamos el vector con la posición de las
alturas ordenadas de menor a mayor. Como hicimos anteriormente
lines(baccharis$altura[ord], p.0[ord], lwd=3, col=4) #
```

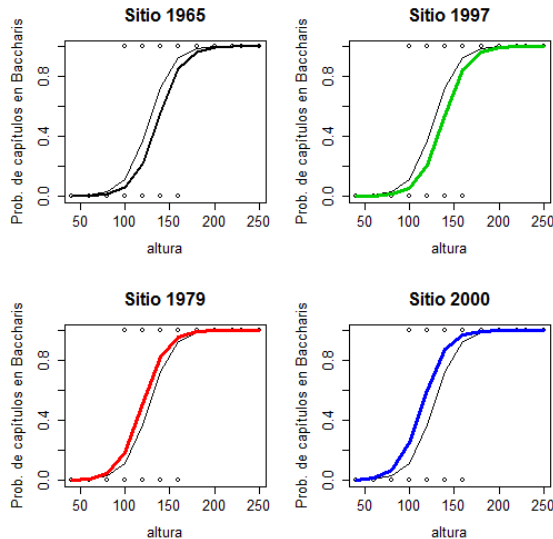


Figura 12. 6: probabilidad de presencia de capítulos en función de la altura de plantas de *Baccharis dracunculifolia* estimada por el modelo "M.0". La línea llena muestra la probabilidad de encontrar un arbusto de baccharis con capítulos en cualquiera de los sitios estimado por el modelo y las líneas de colores, la probabilidad para cada uno de los sitios según la intercepción aleatoria estimada.

### # 9- Modelos glmmADMB

# La librería glmmADMB permite ajustar modelos lineales generalizados de efectos mixtos con distribución Poisson, binomial, binomial negativa, Gamma, Beta, Poisson truncada, zero-inflated (Gaussiana en breve). Para ello, las funciones de enlace que utiliza son: log, logit, probit, cloglog, inverse, identity.  
# Esta librería admite diseños con uno o múltiples efectos aleatorios (anidados o cruzados).

# glmmADMB (a diferencia de lme4) puede ajustar modelos sin efectos aleatorios

# Para utilizar glmmADMB es recomendable estar familiarizado con modelos lineales generalizados mixtos (GLMMs), lo cual requiere conocer (i) modelos lineales generalizados (ej.: regresión logística, binomial, y Poisson) y (ii) modelos mixtos (que maximizan la verosimilitud marginal en lugar de manipular la suma de cuadrados).

# Para ajustar un modelo en glmmADMB debemos:

# - Especificar los efectos fijos del modelo,

# - Especificar los efectos mixtos, siguiendo la notación de nlme y lme4.

# Los efectos aleatorios son especificados como e|g, donde "e" es un efecto y "g" es un factor de agrupamiento (un factor, un anidamiento o, una interacción entre factores). Por ejemplo, la fórmula podría ser 1|bloque para un modelo con intercepción aleatoria o, tiempo|bloque para un modelo con variación aleatoria en las pendientes del tiempo para los grupos especificados por los bloques. Un modelo de efecto de aleatorios anidados (bloques dentro de sitio) sería 1|sitio/bloque; un modelo con efectos aleatorios cruzados (bloque y años) sería (1|bloque)+(1|año).

# - Los efectos aleatorios pueden ser especificados en un argumento aparte (como en nlme) o, como parte del modelo (como en lme4).

```
# - Elegir la distribución del error a través de especificar la familia (ej.: "poisson"
or "binomial")
# - Especificar la función de enlace (ej.: "logit" or "log").
# - De manera opcional , especificar que hay ceros inflados como,
zeroInflation=TRUE (http://glmmadmb.r-forge.r-project.org/glmmADMB.html).
```

```
# install.packages("glmmADMB", repos=c("http://glmmadmb.r-forge.r-
project.org/repos", getOption("repos")),type="source")
```

```
# Otra alternativa
# install.packages("glmmADMB", repos="http://r-forge.r-project.org",
type="source")
```

```
library(glmmADMB)
AMDBmod<-glmmadmb(cap.bi ~ altura +(1|fSitio),
                  family="binomial",
                  zeroInflation=TRUE,
                  data = baccharis)
```

```
summary(AMDBmod)
```

```
Call:
  glmmadmb(formula = cap.bi ~ altura + (1 | fSitio), data = baccharis, family = "binomial", zeroInflation = TRUE)
```

```
AIC: 88.6
```

```
Coefficients:
```

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-9.5584	1.9467	-4.91	9.1e-07	***
altura	0.0752	0.0140	5.37	7.7e-08	***

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

```
Number of observations: total=171, fSitio=4
```

```
Random effect variance(s):
```

```
Group=fSitio
```

	Variance	StdDev
(Intercept)	0.8558	0.9251

```
Zero-inflation: 1e-06 (std. err.: 3.091e-09 )
```

```
Log-likelihood: -40.2807
```

```
# Comparar los parámetros estimados entre el modelo AMDBmod (glmmadmb)
y M.0 (glmer)
```

```
# La librería coefplot2 permite graficar los parámetros ajustados por el modelo
glmmadmb
```

```
library(coefplot2)
```

```
coefplot2(AMDBmod)
```

## # 10- Resumen

```
# Tomamos como ejemplo un experimento que buscaba responder una pregunta
de probabilidad de ocurrencia de un evento. Esta pregunta y la variable
```

respuesta (presencia/ausencia) responder a un caso especial de distribución binomial.

# Generamos modelos con distribución binomial utilizando distintas funciones en R que admiten o no (glm) efectos aleatorios.

# Verificamos la validez de estos modelos mediante el parámetro de sobredispersión "phi".

# Comparamos los resultados de estos modelos.

# Utilizamos los parámetros estimados por el modelo para graficar el modelo general y, a nivel de los sitio (niveles del factor aleatorio de agrupamiento).

### # 11- Tarea

# a- Generar un modelo mixto alternativo

# Por ejemplo, altura como pendiente aleatoria.

# b- Verificar distintas ordenadas y pendientes para cada sitio

# c- Comparar el ajuste entre los modelos

### # 12- Agradecimiento

# Datos provistos por Fernando Biganzoli y William B. Batista de la Facultad de Agronomía de la Universidad de Buenos Aires. Biganzoli, F. 2011. Influencia de los incendios en la dinámica poblacional de dos arbustos dominantes en la sabana mesopotámica. Tesis doctoral, EPG FAUBA. Por razones didácticas y para preservar los trabajos originales, los datos no se presentan de manera completa.



# Modelos lineales generalizados mixtos [GLMM] II

## Contenidos

<b># CAPÍTULO 13: Modelos lineales generalizados mixtos - Conteos.....</b>	<b>210</b>
<b># 1- Configuración inicial.....</b>	<b>210</b>
<b># 2- El caso: Número de granos en cebada.....</b>	<b>210</b>
# 2.a- Pregunta de interés .....	210
# 2.b- Variable respuesta.....	210
# 2.c- Diseño del experimento .....	210
# 2.d- Identificar jerarquías.....	210
<b># 3- Leer el archivo de datos.....</b>	<b>211</b>
<b># 4- Exploremos los datos.....</b>	<b>211</b>
<b># 5- Distribución de los residuos y GLMM .....</b>	<b>211</b>
<b># 6- Modelo GLMM Poisson .....</b>	<b>213</b>
# 6.a- Modelo glmer .....	213
# 6.b- Modelo glmmPQL.....	213
# 6.c- Modelo glmmadmb.....	213
<b># 7- Dispersión del modelo .....</b>	<b>214</b>
# 7.a- Evaluar dispersión I .....	214
# 7.b- Evaluar dispersión II.....	215
# 7.c- Evaluar dispersión III.....	215
<b># 8- Corregir la sobredispersión.....</b>	<b>216</b>
<b># 9- Autocorrelación.....</b>	<b>216</b>
# 9.a- Aclaración para casos con NAs .....	216
# 9.b- ACF.....	217
# 9.c- acf.....	217
<b># 10- Comparamos los modelos .....</b>	<b>219</b>
<b># 11- Verificación del modelo .....</b>	<b>220</b>
# 11.a- Ajustados vs. residuales.....	220
# 11.b- Residuales vs. predictores .....	220
# 11.c- Observados vs. ajustados.....	221
<b># 12- Estimaciones del modelo.....</b>	<b>222</b>
<b># 13- Obtener el <math>R^2</math> del modelo .....</b>	<b>223</b>
<b># 14- Graficar los efectos aleatorios .....</b>	<b>223</b>
<b># 15- Inferencia sobre el componente fijo.....</b>	<b>224</b>
<b># 16- Graficar el modelo general.....</b>	<b>224</b>
<b># 17- Tarea.....</b>	<b>225</b>
<b># 18- Resumen.....</b>	<b>225</b>
<b># 19- Agradecimiento.....</b>	<b>225</b>

## # CAPÍTULO 13: Modelos lineales generalizados mixtos - Conteos

### # 1- Configuración inicial

# Definir el directorio de trabajo  
# setwd("D:\\Mis documentos")

### # 2- El caso: Número de granos en cebada

#### Diseño del experimento

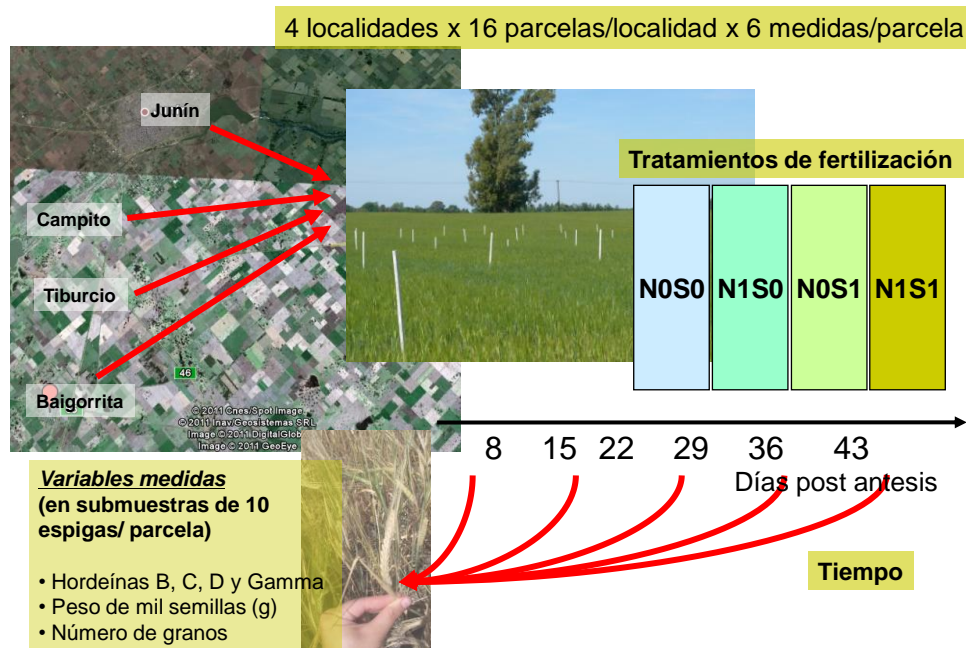


Figura 13. 1: diseño del experimento

#### # 2.a- Pregunta de interés

# ¿Qué impacto tiene la fertilización sobre el número de granos en cebada cervecera?

#### # 2.b- Variable respuesta

# Número de granos (en 10 espigas)

#### # 2.c- Diseño del experimento

# Arreglo factorial de dos factores con dos niveles (Fig. 12.1):

# Fertilización con Nitrógeno: N0 y N4

# Fertilización con Azufre: S0 y S1

# Fechas (4)

#### # 2.d- Identificar jerarquías

# ¿Cuáles son las escalas y qué variables se midieron en cada una?

# Escalas:

# Localidades: Junin y Baigorrita

# Bloques (4) en cada localidad

# Parcelas (4) en cada bloque

# Fechas (4) en cada parcela

```
# Los tratamientos de fertilización fueron asignados a nivel de parcela
# (al inicio del experimento).
# El número de granos fue determinado a nivel de fecha.
```

### # 3- Leer el archivo de datos

```
# En este caso, la tabla de datos "12_Granos.csv"
cebada<-read.csv("12_Granos.csv", header=T, sep=",")
```

```
cebada<-subset(cebada,n_granos>=0)
summary(cebada)
str(cebada)
```

```
# la función "str" devuelve la estructura del objeto:
# los elementos que contiene y la ubicación.
```

```
# IMPORTANTE: dado que hay medidas repetidas en el set de datos,
# es importante ORDENAR la tabla.
# Este paso es fundamental para realizar los cálculos de autocorrelación
empírica (ACF).
cebada<-cebada[with(cebada, order(Localidad, Parcela, Fecha)),]
View(cebada)
# Verificamos que para cada parcela las fechas queden consecutivas.
```

### # 4- Exploremos los datos

#### # Outliers

```
with(cebada, dotchart(n_granos, xlab="Granos en 10 espigas (n)", ylab="Orden
de los datos"))
with(cebada, dotchart(n_granos, groups=Localidad, xlab="Granos en 10 espigas
(n)", ylab="Orden de los datos por localidad"))
with(cebada, dotchart(n_granos, groups=Nitro, xlab="Granos en 10 espigas (n)",
ylab="Orden de los datos por Fert. N"))
with(cebada, dotchart(n_granos, groups=Azufre, xlab="Granos en 10 espigas
(n)", ylab="Orden de los datos por Fet. S"))
# No observamos datos outliers
```

```
with(cebada, boxplot(n_granos~Fecha,ylab="Granos en 10 espigas (n)",
xlab="Fecha"))
```

### # 5- Distribución de los residuos y GLMM

```
# El número de granos es una variable cuantitativa discreta con límite inferior >0
# y límite superior infinito (no sabemos cuál es el valor máximo)
# Este tipo de variable no suele presentar una distribución normal de los
residuos.
# Los residuos suelen cumplir con la distribución poisson o binomial negativa
```

#### # ¿Qué paquetes (y funciones) estiman GLMMs?

```
# Se han desarrollado y se están desarrollando varios paquetes para
# modelos mixtos generalizados.
```

```
#
# lme4::glmer (Laplace approximation and adaptive Gauss-Hermite quadrature
# [AGHQ])
# glmmADMB:glmmadmb (Laplace)
# MCMCglmm (Markov chain Monte Carlo)
# MASS::glmmPQL (penalized quasi-likelihood)

# glmmML (AGHQ)
# glmmAK (AGHQ)
# glmm (from Jim Lindsey's repeated package: AGHQ)
# gamlss.mx
# ASREML-R
# sabreR
# glmmBUGS
# glmmGS
# glmmLasso.

# Para modelos mixtos la función "glmer" del paquete "lme4",
# la función "glmmPQL" del paquete "MASS" y
# la función "glmmadmb" del paquete "glmmADMB"
# permiten indicar el modelo de distribución de probabilidad de los residuos.
# La distribución de probabilidad se indica con el argumento "family".
# Las distribuciones más comunes son: binomial, poisson, gaussian, gamma.
# La función "glmmadmb" admite modelos mixtos con distribución poisson y
# binomial negativa entre otras (Ver práctica 12).
# Algunas funciones admiten también distribuciones quasibinomial y
# quasipoisson.

# "glmmPQL" utiliza las funciones de "lme" y la sintaxis es igual.
# Dado que no utilizan los mismos algoritmos (Laplace y PQL, respectivamente),
# los valores estimados (errores estándar, p-values, etc) por ambos modelos son
# similares pero, no idénticos.

# A diferencia de "glmer" y "glmmadmb", "glmmPQL" no nos devuelve valores de
# ajuste del modelo
# como el AIC, el BIC, el loglikelihood o la deviance.
# En este sentido, con las funciones "glmer" y "glmmadmb" son más fáciles
# comparar modelos.

# Otra diferencia es que "glmer" y "glmmadmb" del paquete "lme4" no admiten
# estructuras de
# correlación como "glmmPQL" al usar las funciones de "lme" del paquete "nlme"

# Es decir, la función "glmmPQL" tiene algoritmos menos robustos (o confiables)
# pero,
# es más flexible en las especificaciones del modelo.
# Por otro lado, la función "glmmadmb" es flexible en el sentido que admite un
# gran número de distribuciones.
```

```
# Otra opción para modelar variables con estas distribuciones es utilizar
modelos bayesianos.
# R posee paquetes que trabajan con "WinBUGs" y otros que permiten realizar
inferencias bayesianas
# http://cran.r-project.org/web/views/Bayesian.html
```

```
# Ben Bolker (2009). GLMM on symbiont effects on coral predation
# Compara varias aproximaciones a los GLMM sobre un mismo set de datos.
# (http://glmm.wdfiles.com/local--files/examples/culcita\_glmm.pdf)
```

### # 6- Modelo GLMM Poisson

```
# En este caso, tenemos un modelo con jerarquías
# (Las fechas se encuentran anidadas en parcelas y, las parcelas anidadas en las
localidades)
# Además, es probable que la función de distribución de los residuales no sea
normal sino Poisson como discutimos anteriormente.
```

```
# Utilizaremos las funciones "glmer", "glmmPQL" y "glmmadmb" que permiten
modelar jerarquías e indicar una distribución poisson (o binomial negativa:
glmmadmb).
```

#### # 6.a- Modelo glmer

```
# Ajustamos el modelo con todos los factores e interacciones
# respetando la estructura jerárquica impuesta por el diseño.
```

```
library(lme4)
cebada$f_fecha=as.factor(cebada$Fecha)
Mod<-glmer(n_granos~Nitro*Azufre*f_fecha
+ (1|Localidad/Parcela), family=poisson(link="log"), data=cebada)
summary(Mod)
```

#### # 6.b- Modelo glmmPQL

```
library(MASS)
ModPQL<-glmmPQL(n_granos~Nitro*Azufre*f_fecha,
random=~1|Localidad/Parcela, family=poisson(link="log"),
data=cebada)
```

```
summary(ModPQL)
```

#### # 6.c- Modelo glmmadmb

```
# install.packages("glmmADMB", repos=c("http://glmmadmb.r-forge.r-
project.org/repos", getOption("repos")),type="source")
```

```
# Alternativa
```

```
# install.packages("glmmADMB", repos="http://r-forge.r-project.org",
type="source")
```

```

library(glmADMB)
vignette("glmADMB", package="glmADMB")

ModADMB_poi<-glmmadmb(n_granos~Nitro*Azufre*f_fecha +
(1|Localidad/Parcela),
  data=cebada,
  zeroInflation=FALSE,
  family="poisson")
# Parameters were estimated, but not standard errors were not
# El modelo lo se ajusta a una distribución "poisson"

# La distribución binomial negativa, es otra distribución estocástica discreta
# Esta distribución es similar a la distribución Poisson pero, a diferencia de la
Poisson, permite heterogeneidad.
# En
# Bolker BM (2008). 4.5 Bestiary of Distributions. En: Ecological models and data
in R. Princenton University Press, Princenton and Oxford. Pages 120-137
# podrán encontrar una descripción y caracterización de las distribuciones más
comunes.

# Ajustamos un modelo con una distribución binomial negativa:
# cuya varianza =  $\mu + \mu^2/k$  o,  $\mu(1 + \mu/k)$  donde, K es el parámetro de
sobredispersión.
ModADMB_nb<-glmmadmb(n_granos~Nitro*Azufre*f_fecha +
(1|Localidad/Parcela),
  data=cebada,
  zeroInflation=FALSE,
  family="nbinom")

# Otra posibilidad es un modelo binomial negativo cuya varianza =  $\mu*k$ . En
este caso, family="nbinom1"
summary(ModADMB_nb)

# 7- Dispersión del modelo
# La distribución poisson asume que la varianza estimada es igual a la media
estimada.
# Este supuesto se verifica mediante el parámetro de dispersión.
# Este parámetro debería ser  $\sim 1$  para que los residuos se distribuyan según una
poisson.
# Cuando es mayor, decimos que hay sobredispersión

# 7.a- Evaluar dispersión I
# La salida de los modelos "glmer" y "glmmPQL" NO incluyen el parámetro de
dispersión.
# Por lo tanto, utilizaremos la función de dispersión propuesta por Bolker
(2009).

```

# Esta función se puede aplicar, al menos, a modelos ajustados por lme4 y glmmADMB (<http://glmm.wikidot.com/faq>)

```
overdisp_fun<- function(model) {
  ## number of variance parameters in an n-by-n variance-covariance matrix
  vpars<- function(m) {
    nrow(m)*(nrow(m)+1)/2
  }
  model.df<- sum(sapply(VarCorr(model),vpars))+length(fixef(model))
  (rdf<- nrow(model@frame)-model.df)
  rp<- residuals(model)
  Pearson.chisq<- sum(rp^2)
  prat<- Pearson.chisq/rdf
  pval<- pchisq(Pearson.chisq, df=rdf, lower.tail=FALSE,log.p=TRUE)
  c(chisq=Pearson.chisq,ratio=prat,p=exp(pval))
}
```

# Una vez que le asignamos la función al objeto, queda en  
# la memoria temporal de R y la utilizamos con el modelo.

```
overdisp_fun(Mod)
#   chisq      ratio      p
# 132.17238076  1.27088828  0.03245969
```

# En este caso, el parámetro de dispersión (ratio) se aleja muy levemente  
# del supuesto del modelo.

### # 7.b- Evaluar dispersión II

# La función "gof" del paquete "aods3" es reciente y se puede aplicar  
# para los modelos ajustados por lme4  
# La función "gof" calcula la deviance "D"  
# y el estadístico chi-squared  $X^2$  de Pearson para el modelo.

```
library(aods3)
gof(Mod)
# D = 1072.082, df = 104, P(>D) = 1.746241e-160
# X2 = 130.273, df = 104, P(>X2) = 0.04155507
```

# Estas funciones: overdisp\_fun y gof, no son aplicables a los modelos glmmPQL (ModPQL) o glmmadmb (ModADMB\_nb)

### # 7.c- Evaluar dispersión III

# La función "glmmadmb" SI nos da el parámetro de sobredispersión.

# Con la sentencia

```
summary(ModADMB_nb)
```

# Aparece casi al pie de la salida

```
# Negative binomial dispersion parameter: 403.43 (std. err.: 0.17297)
```

# Con la sentencia

```
ModADMB_nb
```



# Aparece al principio: es el parámetro "alpha"

```
# GLMM's in R powered by AD Model Builder:
#
# Family: nbinom
# alpha = 403.43
# link = log
```

# A diferencia de la distribución "poisson", la "binomial negativa" es robusta a la sobredispersión.

### # 8- Corregir la sobredispersión

# En caso que deseen (o sea necesario) corregir la sobredispersión  
# dejamos planteado el procedimiento que proponen Bolker et al. (2009) y Gelman y Hill (2007)

```
# Agregamos un factor aleatorio a nivel de observación al modelo.
# Esto permite considerar la sobredispersión.
# Este procedimiento nos da un modelo lognormal-Poisson
# para las muestras individuales (Elton et al. 2001)
Mod2 <- glmer(n_granos~Nitro*Azufre*f_fecha
              + (1|Localidad/Parcela/ID), family=poisson(link="log"), data=cebada)
# Notar que se agrega el ID
```

# IMPORTANTE: generamos adrede un modelo con tantos niveles en el factor aleatorio como observaciones.

```
# Evaluamos la sobredispersión en el nuevo modelo
overdisp_fun(Mod2)
# chisq      ratio      p
# 89.5151880  0.8690795  0.8256163
```

### # 9- Autocorrelación

# Verificar si existe autocorrelación entre las medidas repetidas dentro de la misma parcela

#### # 9.a- Aclaración para casos con NAs

# ¡ATENCIÓN! si la tabla de datos tuviera NAs, debemos generar un nuevo vector de residuales que tenga los NA's en las filas correspondientes.

```
length(cebada$n_granos)
E<-residuals(Mod, type="pearson")
length(E)
```

# Esto es MUY IMPORTANTE para el cálculo de la autocorrelación empírica.  
# De otro modo consideraría que dos residuales contiguos pertenecen a fechas contiguas

# y esto puede no ser así si hubo un NA entre ellos

```
I1 <- !is.na(cebada$n_granos)
```

# Es un vector lógico (TRUE/FALSE) que nos indica las posiciones (filas)



# donde hay NA's

```
Efull<- vector(length = length(cebada$n_granos))
# creamos un vector que tiene el largo de las observaciones.
Efull<- NA # asignamos NA a los valores del vector
Efull[11] <- E # finalmente, asignamos los residuales del modelo (E)
# a las filas donde no había NAs.
# De este modo, reconstruimos el vector de residuales respectando
# la ubicación de los NAs originales de la tabla de datos.
# Con este vector calculamos la autocorrelación.
```

```
acf(Efull, na.action = na.pass, lag.max=3, plot=FALSE, main = "Auto-correlación")
```

# Hay varias funciones para evaluar autocorrelación presentamos  
# dos de las más utilizadas:

### # 9.b- ACF

# La función "ACF" está incluida en el paquete "nlme".  
# En consecuencia, esta implementada para modelos "gls" y "lme".

```
# ACF(object, maxLag, ...)
```

# Los argumentos más importantes de esta función son:  
# object = es el modelo...  
# maxLag = es la máxima cantidad de distancias a la que calcula la autocorrelación.

# IMPORTANTE: esta función no es aplicable a modelos estimados por "glmer" o "glmmadmb" (ambos de "lme4")

```
ACF(ModPQL, maxLag= 3)
```

```
plot(ACF(ModPQL, maxLag= 3))
plot(ACF(ModPQL, maxLag= 3), alpha=0.05)
plot(ACF(ModPQL, maxLag= 3), alpha=0.05, grid=TRUE)
```

# El argumento "alpha" grafica la línea de corte con una probabilidad del 5%  
# El argumento "grid" agrega la grilla cuadrangular.

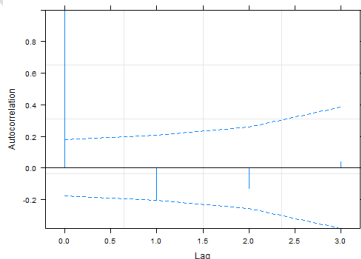


Figura 13. 2: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "ModPQL". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.

### # 9.c- acf

# La función "acf" en el paquete "stats" es similar a la función ACF

# Esta función es aplicable a modelos "glmer" u otros del paquete "lme4"

```
# acf(x, ....)
# x= son los residuales del modelo,
# lag.max = es la máxima cantidad de distancias a la que calcula la
autocorrelación,
# type = c("correlation", "covariance", "partial"),
# plot = TRUE (o, FALSE devuelve los valores de autocorrelación no graficados),
# na.action = na.fail, ...)
```

```
layout(matrix(1:4, 2, 2))
```

```
acf(residuals(Mod), lag.max=3, plot=FALSE)
acf(residuals(Mod, type="pearson"), lag.max=3, plot=FALSE)
acf(residuals(Mod, type="pearson"), lag.max=3, plot=TRUE)
```

# Con la función glmer, el type en residuals, cambia...ya no admite "normalized", sino que debería ser "pearson".

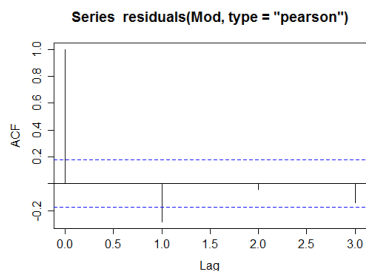


Figura 13. 3: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Mod". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones

```
# Lo comparamos con el modelo ajustado por glmmPQL
acf(residuals(ModPQL, type="normalized"), lag.max=3, plot=FALSE)
acf(residuals(ModPQL, type="normalized"), lag.max=3, plot=TRUE)
```

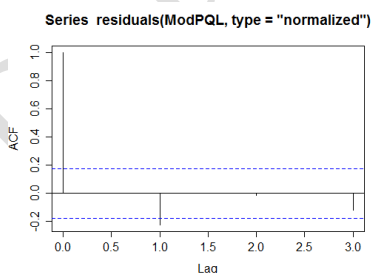


Figura 13. 4: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "ModPQL". Las líneas llenas veticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.

```
# con el modelo ajustado por glmmadmb
acf(residuals(ModADMB_nb, type="pearson"), lag.max=3, plot=TRUE)
```

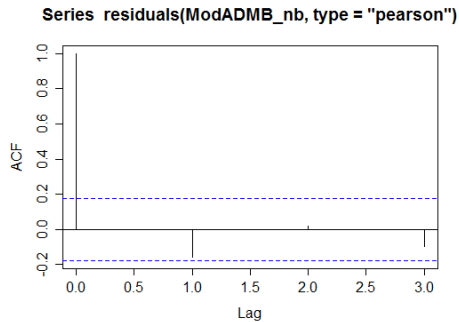


Figura 13. 5: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "ModADMB". Las líneas llenas veticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.

# con el modelo ajustado por glmer corrigiendo sobredispersión  
`acf(residuals(Mod2, type="pearson"), lag.max=3, plot=TRUE)`

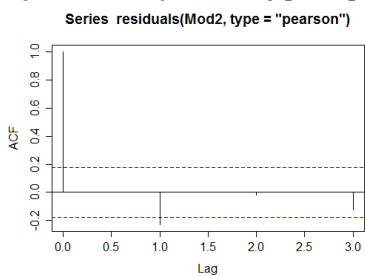


Figura 13. 6: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Mod2". Las líneas llenas veticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.

# Podemos considerar que los niveles de auto-correlación no son importantes en ninguno de los casos.

### # 10- Comparamos los modelos

# OBSERVAR los valores de ajuste de los modelos

`AIC(Mod)`

# [1] 1108.1

`AIC(ModADMB_nb)`

# [1] 1109.3

`AIC(Mod2)`

# [1] 1107.578

# En cambio, no tenemos un valor de AIC para el modelo ModPQL.

# En consecuencia, tampoco podemos incluirlo en la comparación realizada con la función AICtab

`library("bbmle")`

`AICtab(Mod, ModADMB_nb, Mod2, weights = T, delta = TRUE, base=T, sort = TRUE)`

	AIC	dAIC	df	weight
# Mod2	1107.6	0.0	19	0.45
# Mod	1108.1	0.5	18	0.35
# ModADMB_nb	1109.3	1.7	19	0.19

# Observar que la diferencia de ajuste entre los tres modelos no es muy importante.

# El delta AIC del último modelo es de 1.7

## # 11- Verificación del modelo

### # 11.a- Ajustados vs. residuales

```
Fit<- fitted(Mod)
```

```
E<-residuals(Mod, type="pearson")
```

```
FitPQL<- fitted(ModPQL)
```

```
EPQL<-residuals(ModPQL, type="normalized")
```

```
FitADMB<- fitted(ModADMB_nb)
```

```
EADMB<-residuals(ModADMB_nb, type="pearson")
```

```
Fit2<- fitted(Mod2)
```

```
E2<-residuals(Mod2, type="pearson")
```

```
layout(matrix(1:4, 2, 2))
```

```
plot(x=Fit, y=E, xlab="Ajustados", ylab="Residuales", main="modelo glmer")
```

```
abline(0,0, col="red", lwd=3)
```

```
plot(x=FitPQL, y=EPQL, xlab="Ajustados", ylab="Residuales", main="modelo  
glmmPQL")
```

```
abline(0,0, col="red", lwd=3)
```

```
plot(x=FitADMB, y=EADMB, xlab="Ajustados", ylab="Residuales", main="modelo  
glmmadmb")
```

```
abline(0,0, col="red", lwd=3)
```

```
plot(x=Fit2, y=E2, xlab="Ajustados", ylab="Residuales", main="modelo glmer2")
```

```
abline(0,0, col="red", lwd=3)
```

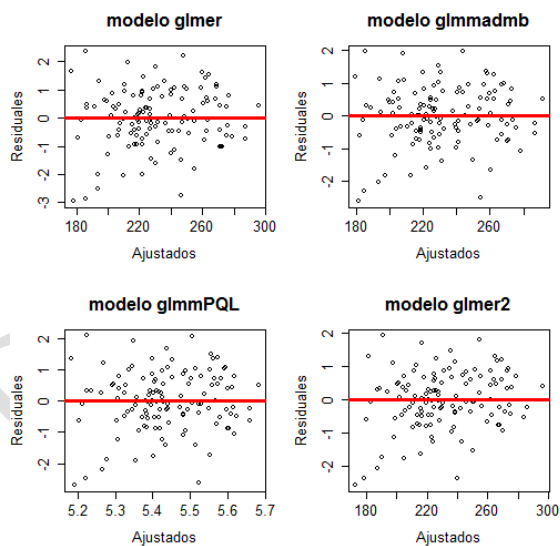


Figura 13. 7: residuales normalizados (o "pearson") en relación a los valores ajustados por los modelos glmer (Mod, family="poisson"), glmmPQL (ModPQL, family="poisson"), glmmADMB (ModADMB\_nb, family="nbinom") y glmer [Mod2, family="poisson" con corrección para sobredispersión según Bolker et al. (2009) y Gelman y Hill (2007)]. La línea roja indica residual cero. Observar que la escala en el eje de las ordenadas no es idéntico entre gráficos.

### # 11.b- Residuales vs. predictores

```
layout(matrix(1:4, 2, 2))
```

```
par(mfrow = c(2, 2))
```

**# Modelo glmer**

```
boxplot(E~cebada$Localidad, main="Localidad", ylab="Residuales",  
xlab="Localidad")  
boxplot(E~cebada$Nitro, main="Nitrogeno", ylab="Residuales",  
xlab="Nitrogeno")  
boxplot(E~cebada$Azufre, main="Azufre", ylab="Residuales", xlab="Azufre")  
boxplot(E~cebada$f_fecha, main="Fecha", ylab="Residuales", xlab="Fechas")
```

**# Modelo glmmPQL**

```
boxplot(EPQL~cebada$Localidad, main="Localidad", ylab="Residuales",  
xlab="Localidad")  
boxplot(EPQL~cebada$Nitro, main="Nitrogeno", ylab="Residuales",  
xlab="Nitrogeno")  
boxplot(EPQL~cebada$Azufre, main="Azufre", ylab="Residuales",  
xlab="Azufre")  
boxplot(EPQL~cebada$f_fecha, main="Fecha", ylab="Residuales",  
xlab="Fechas")
```

**# Modelo glmmADMB**

```
boxplot(EADMB~cebada$Localidad, main="Localidad", ylab="Residuales",  
xlab="Localidad")  
boxplot(EADMB~cebada$Nitro, main="Nitrogeno", ylab="Residuales",  
xlab="Nitrogeno")  
boxplot(EADMB~cebada$Azufre, main="Azufre", ylab="Residuales",  
xlab="Azufre")  
boxplot(EADMB~cebada$f_fecha, main="Fecha", ylab="Residuales",  
xlab="Fechas")
```

**# Modelo glmer2**

```
boxplot(E2~cebada$Localidad, main="Localidad", ylab="Residuales",  
xlab="Localidad")  
boxplot(E2~cebada$Nitro, main="Nitrogeno", ylab="Residuales",  
xlab="Nitrogeno")  
boxplot(E2~cebada$Azufre, main="Azufre", ylab="Residuales", xlab="Azufre")  
boxplot(E2~cebada$f_fecha, main="Fecha", ylab="Residuales", xlab="Fechas")
```

**# 11.c- Observados vs. ajustados**

```
layout(matrix(1:4, 2,2))  
plot(cebada$n_granos~Fit, xlim=c(120,300), ylim=c(120,300), main="Modelo  
glmer", ylab="Observados", xlab="Predichos")  
abline(0,1, col="red")  
  
plot(cebada$n_granos~exp(FitPQL), xlim=c(120,300), ylim=c(120,300),  
main="Modelos glmmPQL", ylab="Observados", xlab="Predichos")  
abline(0,1, col="red")
```

```
plot(cebada$n_granos~FitADMB, xlim=c(120,300), ylim=c(120,300),
main="Modelos glmmADMB", ylab="Observados", xlab="Predichos")
abline(0,1, col="red")
```

```
plot(cebada$n_granos~Fit2, xlim=c(120,300), ylim=c(120,300), main="Modelo
glmer2", ylab="Observados", xlab="Predichos")
abline(0,1, col="red")
```

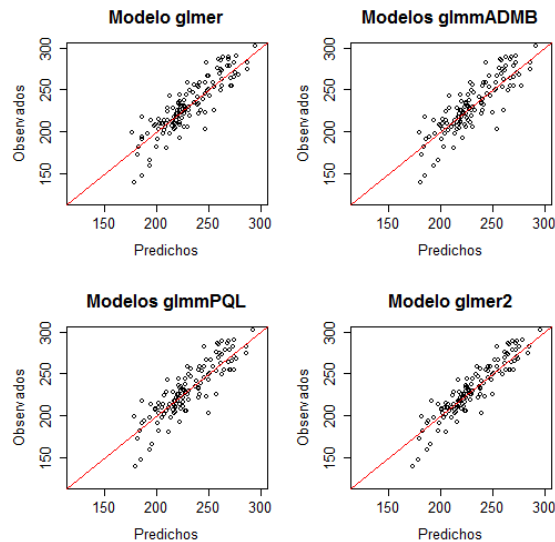


Figura 13. 8: valores observados de número de granos en relación a los valores ajustados por los modelos glmer (Mod, family="poisson"), glmmPQL (ModPQL, family="poisson"), glmmADMB (ModADMB\_nb, family="nbinom") y glmer [Mod2, family="poisson" con corrección para sobredispersión según Bolker et al. (2009) y Gelman y Hill (2007)]. La línea roja indica la relación 1:1.

# Los gráficos no muestran patrones importantes en la distribución de los residuales.

# Podemos considerar que los modelos son adecuados.

### # 12- Estimaciones del modelo

# Observar qué le podemos pedir a cada uno

```
names(attributes(Mod))
```

```
names(ModPQL)
```

```
names(ModADMB_nb)
```

### # Comparar los efectos fijos estimados por los modelos

```
fixef(Mod)
```

```
fixef(ModPQL)
```

```
ModPQL$coefficients$fixed
```

# Son dos formas equivalentes para mostrar los valores estimados para los factores fijos del modelo.

```
fixef(ModADMB_nb)
```

```
fixef(Mod2)
```

*# Comparar los efectos aleatorios estimados por los modelos*

```
ranef(Mod)$Localidad
ranef(ModPQL)$Localidad
ranef(ModADMB_nb)$Localidad
ranef(Mod2)$Localidad
```

**# 13- Obtener el R<sup>2</sup> del modelo**

# Nakagawa et al. (2013) proponen una manera simple de obtener el R<sup>2</sup> de un modelo de efectos mixtos.

# El trabajo muestra ejemplos para los casos de distribución normal, binomial y poisson ajustados con la función glmer (lme4).

# Seguimos la propuesta de Nakagawa et al. (2013) para el modelo Mod2

# Ajustamos el modelo nulo, sin los efectos fijos pero, incluyendo todos los efectos aleatorios

# Al igual que para Mod2, ID nos permite estimar la dispersión aditiva en lmer y nos generará un mensaje de error o alerta al correr el modelo

```
Mod2Null <- glmer(n_granos~1 + (1|Localidad/Parcela/ID),
family=poisson(link="log"), data=cebada)
```

# Calculamos la varianza de los valores ajustados

```
VarF <- var(as.vector(fixef(Mod2) %*% t(Mod2@pp$X)))
```

*# R<sup>2</sup> marginal*

# El R<sup>2</sup> marginal describe la proporción de varianza explicada por el factor/es fijo/s.

# fixef(Mod2Null) devuelve los estimados para la intercepción del modelo nulo

```
VarF/(VarF + VarCorr(Mod2)$Localidad[1] + VarCorr(Mod2)$Parcela[1] +
VarCorr(Mod2)$ID[1] + log(1 + 1/exp(as.numeric(fixef(Mod2Null)))))
[1] 0.3032769
```

*# R<sup>2</sup> condicional*

# El R<sup>2</sup> condicional describe la proporción de la varianza explicada tanto por los factores fijos como aleatorios.

```
(VarF + VarCorr(Mod2)$Localidad[1] + VarCorr(Mod2)$Parcela[1])/(VarF +
VarCorr(Mod2)$Localidad[1] + VarCorr(Mod2)$Parcela[1] +
VarCorr(Mod2)$ID[1] + log(1 + 1/exp(as.numeric(fixef(Mod2Null)))))
[1] 0.721846
```

**# 14- Graficar los efectos aleatorios**

# Podemos pedirle y graficar solo los valores estimados a nivel de Localidad

```
plot(ranef(ModPQL),level=1)
```

### # 15- Inferencia sobre el componente fijo

# Utilizamos el modelo ModADMB\_nb (consideramos que no hay diferencias entre los modelos y este posee el mejor gráfico de acf)

```
library(car)
```

```
Anova(ModADMB_nb, test.statistic="Chisq")
```

```
# Analysis of Deviance Table (Type II tests)
```

```
#
```

```
# Response: n_granos
```

	Df	Chisq	Pr(>Chisq)
# Nitro	1	0.4659	0.49489
# Azufre	1	1.1660	0.28022
# f_fecha	3	22.6068	4.877e-05 ***
# Nitro:Azufre	1	1.0634	0.30245
# Nitro:f_fecha	3	1.7726	0.62092
# Azufre:f_fecha	3	7.7451	0.05158 .
# Nitro:Azufre:f_fecha	3	1.1134	0.77383
# Residuals	103		

```
# ---
```

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

### # 16- Graficar el modelo general

```
layout(matrix(1:1,1,1))
```

```
with(cebada,plot(x=Fecha, y = n_granos,
                type = "p",lwd=2 ,col="gray",
                ylab=" Granos (n)", xlab="Fecha", main=""))
```

```
cl<-with(cebada, length(levels(Nitro:Azufre)))
```

```
cebada$trat<-interaction(cebada$Nitro,cebada$Azufre)
```

### # Valores observados

#### ## Nitrogeno N0 + Azufre S0

```
points(cebada$Fecha[cebada$trat == "N0.S0"],cebada$n_granos[cebada$trat ==
"N0.S0"],
```

```
      col = "black", pch = 16)
```

```
legend("bottomright",levels(cebada$trat), col=1:cl, pch=16, bty="p")
```

#### ## Nitrogeno N0 + Azufre S1

```
points(cebada$Fecha[cebada$trat == "N0.S1"],cebada$n_granos[cebada$trat ==
"N0.S1"],
```

```
      col = "green", pch = 16)
```

#### ## Nitrogeno N1 + Azufre S0

```
points(cebada$Fecha[cebada$trat == "N1.S0"],cebada$n_granos[cebada$trat ==
"N1.S0"],
```

```
      col = "red", pch = 16)
```

#### ## Nitrogeno N1 + Azufre S1

```
points(cebada$Fecha[cebada$trat == "N1.S1"],cebada$n_granos[cebada$trat ==
"N1.S1"],
```



```
col = "blue", pch = 16)
```

### # 17- Tarea

# Agregar puntos que indican los estimados del modelo general y para cada localidad utilizando las sentencias de la práctica 5, líneas 461 y siguientes

# Agregar los efectos aleatorios a nivel de localidad

# Efectos aleatorios a nivel de localidad

```
ranef(ModADMB_nb, condVar=T)$Localidad
```

```
#      (Intercept)
```

```
# baigo  0.08349916
```

```
# junín -0.08333040
```

```
baigo<-ranef(ModADMB_nb, condVar=T)$Localidad[[1,1]]
```

# IMPORTANTE: debemos retro -transformar los estimado con "exp".

# Recordamos que la distribución poisson utiliza la función "log".

# Se puede repetir la figura a partir del modelo glmer y comparar

### # 18- Resumen

# Generamos modelos con distribución poisson y binomial negativa utilizando distintas funciones en R que admiten este tipo de distribución.

# Verificamos la validez de estos modelos mediante el parámetro de sobre dispersión "phi".

# Comparamos los resultados de estos modelos.

# Utilizamos los parámetros estimados por el modelo para graficar el modelo general y, a nivel de los sitio (niveles del factor aleatorio de agrupamiento).

### # 19- Agradecimiento

# Datos gentilmente provistos por Andrés Peton y Eduardo Pagano de la Facultad de Agronomía de la Universidad de Buenos Aires. Peton, A. "Patrón de acumulación de hordeínas en granos de cebada cervecera y su relación con la fertilización azufrada y nitrogenada". Tesis de maestría en curso, EPG-FAUBA. Por razones didácticas, los datos presentados son un subconjunto modificado de los datos originales.

# Ejercicio

---

## Contenidos

# CAPÍTULO 14: Ejercicio .....	227
# 1- Configuración inicial.....	227
# 1.a- Definir el directorio de trabajo .....	227
# 1.b- Leer el archivo de datos .....	227
# 2- El caso: Recuperación del forraje luego del pastoreo.....	227
# 3- Para el experimento presentado.....	227

---

---

## # CAPÍTULO 14: Ejercicio

### # 1- Configuración inicial

#### # 1.a- Definir el directorio de trabajo

```
# setwd("D:\\Mis documentos")
```

#### # 1.b- Leer el archivo de datos

```
Leemos el archivo "14_Mas_practica.csv"
```

```
# practica<-.....
```

### # 2- El caso: Recuperación del forraje luego del pastoreo

Para lograr una planificación eficiente y sustentable del uso de los recursos en los sistemas ganaderos es importante conocer de qué factores depende la recuperación (crecimiento) del forraje luego del pastoreo. Para ello se realizó un experimento que consideró el origen del recurso forrajero (Pastizal o Pastura), la intensidad de consumo. El experimento se realizó en tres localidades donde se identificaron 8 parcelas (4 en pastizales naturales y 4 en pasturas). A su vez, en cada parcela se realizaron mediciones en tres estaciones del año. Se midió el consumo por el ganado ( $\text{kg ha}^{-1}$ ) y el crecimiento de forraje ( $\text{kg ha}^{-1}$ ) luego de 30 días de retirar los animales de la parcela.

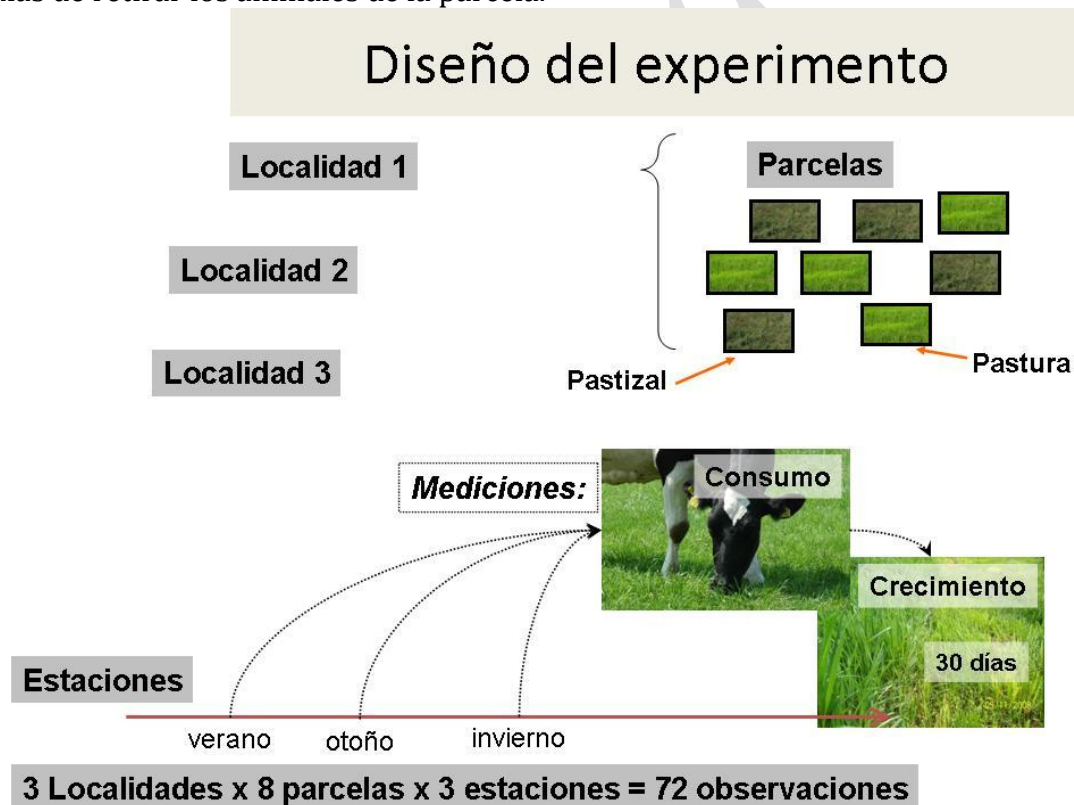


Figura 14. 1: diseño del experimento

### # 3- Para el experimento presentado

- 1) Identificar las unidades experimentales para cada nivel jerárquico.
- 2) Identificar la variable respuesta.

- 3) Identificar las variables predictoras para cada nivel e indicar si son categóricas (factores) o cuantitativas.
- 4) Escribir al menos un modelo adecuado que conteste a la pregunta de interés según la nomenclatura Gellman y Hill (2007) discutida en el curso.
- 5) Escribir en R las sentencias necesarias para realizar algún gráfico exploratorio de los datos que sea interesante en el contexto de la pregunta.
- 6) Escribir en R las sentencias necesarias para plantear el modelo definido en el punto 4).
- 7) Plantear algún gráfico para evaluar los supuestos del modelo. Indicar cuáles son estos supuestos.
- 8) Indicar e interpretar el valor estimado de algún efecto fijo.
- 9) Identificar el parámetro que indica la bondad de ajuste.
- 10) Identificar y explicar los parámetros de la partición de la varianza no explicada por el modelo.

# Índice de funciones

abline, 37  
ACF, 112, 116  
aggregate, 44  
AIC, 70  
AICtab, 71, 87, 114, 174  
anova, 49, 50, 58, 69, 76  
Anova, 54  
arrows, 168  
as.factor, 161  
barplot, 60, 140, 168  
boxplot, 22, 37, 54  
bubble, 152, 177  
cbind, 44, 185  
coef, 201, 205  
colnames, 44  
confint, 202  
coordinates, 152  
coplot, 30  
cor, 26, 125  
curve, 99  
data.frame, 52, 152  
det, 27  
dim, 13, 17  
dotchart, 22, 64  
dredge, 59, 139, 182  
exp, 204  
fitted, 36, 47, 54  
fixef, 98, 99, 201  
formula, 111, 161  
get.models, 141, 182, 183  
getwd, 9  
glht, 77, 167  
glmer, 198  
glmmadmb, 207  
gls, 36, 152  
head, 13, 59, 140  
histogram, 24  
ifelse, 195  
importance, 60, 183  
interaction, 99  
intervals, 67  
inv.logit, 204  
layout, 36, 54  
legend, 77, 98  
length, 12, 97, 110  
levels, 97  
library, 77, 83  
lines, 204  
lm, 45, 125  
lme, 36, 46, 47, 49, 65, 70  
lme.formula, 73  
lmeControl, 164  
lmer, 35, 174  
matrix, 36, 54  
max, 17  
mean, 12, 16  
median, 12, 16  
model.sel, 139, 182  
na.omit, 18  
names, 13, 56, 75  
nrow, 59  
order, 126, 204  
overdisp\_fun, 198  
pairs, 28, 125, 173  
par, 189  
paste, 57  
pchisq, 69  
plot, 26, 37, 54, 77  
plot3d, 151  
points, 77, 100  
qqline, 68, 72  
qqnorm, 68, 72  
ranef, 99, 201  
read.csv, 12, 15, 109  
read.delim, 12, 160  
read.table, 12  
rep, 44  
resid, 36, 47, 54  
round, 26  
scale, 185  
scatterplot3d, 191  
sd, 12, 16  
se.coef, 203  
setwd, 15  
str, 13, 211  
subset, 17, 110, 150, 205

sum, 197  
summary, 13, 15, 16, 37, 49, 73  
table, 16, 26  
tapply, 16, 68  
title, 150  
update, 28, 49, 50, 59, 76, 114, 179  
variogram, 153  
Variogram, 162  
vcov, 199  
View, 12, 126, 161  
vif, 27  
which, 106  
which.max, 17, 28  
wireframe, 151  
with, 22  
write.csv, 18  
write.table, 140, 182  
xyplot, 82

Datos jerárquicos

# Índice de figuras

## Capítulo 2

Figura 2. 1: esquema del diseño experimental que evalúa la producción de seda de dos líneas de gusanos de seda.....	21
Figura 2. 2: boxplot de la distribución de los valores de peso de la corteza (g/gusano) de los gusanos de seda.....	22
Figura 2. 3: distribución de los valores de peso de la corteza (g/gusano) de los gusanos de seda.....	23
Figura 2. 4: distribución de los valores de peso de la corteza (g/gusano) de los gusanos de seda agrupados por línea genética (EC o Eoro). ....	23
Figura 2. 5: boxplot de los valores de peso de la corteza (g/gusano) de los gusanos de seda agrupados por línea genética (EC o Eoro).....	24
Figura 2. 6: boxplot de los valores de peso de la corteza (g/gusano) de los gusanos de seda agrupados por caja (A, B, C, D, E, o F).....	24
Figura 2.7: histograma de la distribución del peso de la corteza (g/gusano) de los gusanos de seda agrupados por línea genética (EC o Eoro). ....	25
Figura 2. 8: histograma de la distribución del peso de la corteza (g/ gusano) de los gusanos de seda agrupados por línea cajas (A, B, C, D, F).....	25
Figura 2. 9: frecuencia de valores del peso de la corteza (g/ gusano) de los gusanos de seda..	26
Figura 2. 10: relación entre variable cuantitativas: peso de las pupas, peso de las larvas y peso de los capullos (todas expresadas en g/gusano de seda).....	28
Figura 2. 11: relación entre variable cuantitativas: peso de las pupas, peso de las larvas y peso de los capullos (todas expresadas en g/gusano de seda). Distribución de los puntos en el triángulo superior de la matriz y valor de coeficiente de Pearson en el triángulo inferior de la matriz.....	29
Figura 2. 12: gráfico de coplot que relaciona el paso de la corteza y de las larvas (expresados en g/gusano de seda) para cada una de las seis cajas.....	30

## Capítulo 3

Figura 3. 1: esquema del diseño experimental que evalúa la producción de seda de dos líneas de gusanos de seda.....	33
Figura 3. 2: residuales normalizados del modelo "mod_lmer" en relación a los valores ajustados por el modelo (izq.), de las líneas genéticas (centro) y del genero (der.). ....	37

## Capítulo 4

Figura 4. 1: esquema del diseño experimental que evalúa la producción de seda de dos líneas de gusanos de seda.....	<b>¡Error! Marcador no definido.</b>
Figura 4. 2: residuales normalizados del modelo "p.p" en relación a los valores ajustados por el modelo y cada uno de los predictores (genero de los gusanos, peso de las pupas, linea genética y peso de las larvas). ....	<b>¡Error! Marcador no definido.</b>
Figura 4. 3: residuales normalizados del modelo "p.p19" en relación a los valores ajustados por el modelo, cada uno de los predictores (género, peso de las pupas, linea genética y peso de	

las larvas). Valores ajustados por el modelo en relación a los observados en el cuadrante inferior derecho. .... **¡Error! Marcador no definido.**

### Capítulo 5

Figura 5. 1: diseño del experimento ..... **¡Error! Marcador no definido.**

Figura 5. 2: residuales normalizados del modelo " M.lme1" en relación a los niveles de carga animal, de la calidad del sitio (Pastizal o Pastura) y, de la interacción entre ambos. .... **¡Error! Marcador no definido.**

Figura 5. 3: residuales normalizados de los modelos "M.lme1" (Izq.) y "M.lme2" (Der.) en relación a los niveles de carga animal..... **¡Error! Marcador no definido.**

Figura 5. 4: gráficos de normalidad para los modelos "M.lme1" (Izq.) y "M.lme2" (Der.). **¡Error! Marcador no definido.**

Figura 5. 5: residuales normalizados en relación a los valores ajustados para los modelos "M.lme1" (izq.) y "M.lme2" (der.)..... **¡Error! Marcador no definido.**

Figura 5. 6: residuales normalizados en relación a la calidad del sitio (Pastizal o Pastura. Izq.) y, la interacción entre nivel de carga x calidad de sitio para los modelos "M.lme3" y "M.lme4", respectivamente. Dertalles de los modelos están descriptos en el cuerpo del texto. .... **¡Error! Marcador no definido.**

Figura 5. 7: residuales normalizados en relación a los valores ajustados para los modelos "M.lme3" (Izq.) y "M.lme4" (Der.). Dertalles de los modelos están descriptos en el cuerpo del texto..... **¡Error! Marcador no definido.**

Figura 5. 8: residuales normalizados del modelo "M.lme4" en relación a los valores ajustados por el modelo, la calidad del sitio y, el nivel de carga. Valores ajustados por el modelo en relación a los observados en el cuadrante inferior derecho. .... **¡Error! Marcador no definido.**

Figura 5. 9: gráficos de normalidad para los modelos "M.lme4" (izq.) y "M.lme1" (der.). **¡Error! Marcador no definido.**

Figura 5. 10: eficiencia de consumo en función del nivel de cara animal. Los puntos azules muestran los valores ajustados por el modelo general. Los puntos rojo y verde para los sitios (aleatorios) de Pastizal y Pastura, respectivamente. .... **¡Error! Marcador no definido.**

### Capítulo 6

Figura 6. 1: esquema del diseño experimental. En este capítulo utilizaremos solo los datos correspondientes a los 36 días post antesis. .... 81

Figura 6. 2: residuales normalizados del modelo " mod " en relación a los valores ajustados por el modelo, la Localidad, en nivel de fertilización con nitrógeno (N), el nivel de fertilización con azufre (S), la interacción N x S y, el peso de mil semillas. .... 84

Figura 6. 3: residuales normalizados en relación a la Localidad para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad ( $\text{VarIdent}(\text{Localidad})$ ), c) con una función que corrige para la interacción N\*S ( $\text{VarIdent}(N*S)$ ) y, d) con una función exponencial para el peso de mil semillas ( $\text{VarExp}(\text{PMS})$ ). .... 88

Figura 6. 4: residuales normalizados en relación a la interacción N\*S para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad ( $\text{VarIdent}(\text{Localidad})$ ), c) con una función que corrige para la interacción N\*S ( $\text{VarIdent}(N*S)$ ) y, d) con una función exponencial para el peso de mil semillas ( $\text{VarExp}(\text{PMS})$ ). .... 89

Figura 6. 5: residuales normalizados en relación a la interacción N\*S para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad ( $\text{VarIdent}(\text{Localidad})$ ), c) con una función que corrige para la interacción N\*S ( $\text{VarIdent}(N*S)$ ) y, d) con una función exponencial para el peso de mil semillas ( $\text{VarExp}(\text{PMS})$ ). .... 90



Figura 6. 6: gráficos de normalidad para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad (VarIdent(Localidad), c) con una función que corrige para la interacción N*S (VarIdent(N*S) y, d) con una función exponencial para el peso de mil semillas (VarExp(PMS)).	91
Figura 6. 7: residuales normalizados en relación a los valores ajustados para modelos lineales mixtos (ajustados por lme) que varían solo en la función de la varianza. El modelo a) inicial, sin función de la varianza, b) con una función que corrige varianzas a nivel de Localidad (VarIdent(Localidad), c) con una función que corrige para la interacción N*S (VarIdent(N*S) y, d) con una función exponencial para el peso de mil semillas (VarExp(PMS)).	92
Figura 6. 8: residuales normalizados del modelo "modFinal" en relación a los valores ajustados por el modelo, cada uno de los predictores (fertilización con azufre, fertilización con nitrógeno y peso de mil semillas).	97
Figura 6. 9: valores ajustados por el modelo "modFinal" en relación a los valores de hordeínas B observados.	97
Figura 6. 10: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g) y ajuste general del modelo (líneas llenas). Los colores indican los tratamientos NconScon (negro), NsinScon (rojo), NconSsin (verde), NsinSsin (azul).	99
Figura 6. 11: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g), ajuste general del modelo (líneas punteadas) y, ajuste considerando la intercepción aleatoria para la localidad de Baigorrita (líneas llenas). Los colores indican los tratamientos NconScon (negro), NsinScon (rojo), NconSsin (verde), NsinSsin (azul).	101
Figura 6. 12: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g), ajuste general del modelo (líneas punteadas) y, ajuste considerando la intercepción aleatoria para la localidad de Campito (líneas llenas). Los colores indican los tratamientos NconScon (negro), NsinScon (rojo), NconSsin (verde), NsinSsin (azul).	103
Figura 6. 13: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g), ajuste general del modelo (líneas punteadas) y, ajuste considerando la intercepción aleatoria para la localidad de Junín (líneas llenas). Los colores indican los tratamientos NconScon (negro), NsinScon (rojo), NconSsin (verde), NsinSsin (azul).	105
Figura 6. 14: cantidad de hordeínas observadas (puntos) en función del peso de mill semillas (g), ajuste general del modelo (líneas punteadas) y, ajuste considerando la intercepción aleatoria para la localidad de Tiburcio (líneas llenas). Los colores indican los tratamientos NconScon (negro), NconSsin (verde).	106

## Capítulo 7

Figura 7. 1: esquema del experimento.	109
Figura 7. 2: alturas (cm) observadas en el tiempo (Fechas de 1 a 5) de las plantas de <i>Baccharis medullosa</i> (puntos negros) y <i>Eupatorium buniifolium</i> (puntos rojos).	110
Figura 7. 3: alturas (cm) observadas en el tiempo (Fechas de 1 a 5) de las plantas de <i>Baccharis medullosa</i> .	111
Figura 7. 4: residuales normalizados del modelo "Bm_mod" en relación a los valores ajustados por el modelo y cada uno de los predictores (Sustrato, riego y fecha).	112
Figura 7. 5: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Bm_mod". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.	113
Figura 7. 6: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Bm_mod". Las líneas llenas veticales indican el nivel de significancia y las líneas	

punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.....	114
Figura 7. 7: residuales normalizados del modelo "Bm_mod12" en relación a los valores ajustados por el modelo y cada uno de los predictores (Sustrato, Riego e interacción Sustrato x Riego).....	115
Figura 7. 8: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Bm_mod12". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.....	116
Figura 7. 9: residuales normalizados del modelo "Er12.ve" en relación a los valores ajustados por el modelo y cada uno de los predictores (Sustrato, Riego e interacción Sustrato x Riego). .....	117
Figura 7. 10: altura de las plantas de <i>Baccharis medullosa</i> (cm) en el tiempo (fechas) y ajuste general del modelo (líneas llenas). Los colores indican los tratamientos: r.a (riego y arena. negro), s.a (sin riego y arena. rojo), r.t (riego y tierra. verde), s.t (sin riego y tierra. azul).....	119

## Capítulo 8

Figura 8. 1: diseño del experimento .....	122
Figura 8. 2: exploración de los datos. Valores de hordeínas B en relación a los predictores (de izquierda superior a derecha inferior): niveles de fertilización con nitrógeno, niveles de fertilización con azufre, la interacción nitrógeno x azufre, las localidades, el peso de mil semillas y los días post antesis de <i>Hordeum vulgare</i> .....	124
Figura 8. 3: relación entre variable cuantitativas: peso de mil semillas (g) y días post antesis (dpa) de <i>Hordeum vulgare</i> .....	125
Figura 8. 4: residuales normalizados del modelo "mod" en relación a los valores ajustados por el modelo y cada uno de los predictores (Localidad, Parcela, Fertilización con nitrógeno, Fertilización con azufre y días post antesis. De izquierda a derecha).....	127
Figura 8. 5: residuales normalizados en relación a los valores ajustados para los modelos lineal (mod. Izq.), logarítmico [ $\log(\text{dpa})$ . mod2. Centro] y, cuadrático [ $\text{dpa} + \text{dpa}^2$ . mod3. Der.]. La líneas rojas indican los valores de cero para los residuales (horizontal) y los predichos (vertical). .....	129
Figura 8. 6: residuales normalizados en relación a los días post antesis (dpa) para los modelos lineal (mod. Izq.), logarítmico [ $\log(\text{dpa})$ . mod2. Centro] y, cuadrático [ $\text{dpa} + \text{dpa}^2$ . mod3. Der.]. .....	130
Figura 8. 7: gráfico de normalidad (cuantiles teóricos) para los modelos lineal (mod. Izq.), logarítmico [ $\log(\text{dpa})$ . mod2. Centro] y, cuadrático [ $\text{dpa} + \text{dpa}^2$ . mod3. Der.]. .....	130
Figura 8. 8: residuales normalizados del modelo "mod3" en relación a los predictores: Localidad, Fertilización con nitrógeno, Fertilización con azufre, la interacción N*S y días post antesis. De izquierda a derecha.....	131
Figura 8. 9: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "mod3". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.....	132
Figura 8. 10: residuales normalizados del modelo "mod.ARMA30" en relación a los valores ajustados por el modelo y cada uno de los predictores: Localidad, Parcela, Fertilización con nitrógeno, Fertilización con azufre y días post antesis. De izquierda a derecha. Las líneas rojas indican valores de cero para los residuales (horizontal) y los ajustados (vertical). .....	135
Figura 8. 11: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para los modelos "mod8" (Izq.), "mod.ARMA30" (Centro) y "mod3" (Der.). Las líneas llenas	

verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.....	136
Figura 8. 12: residuales normalizados en relación a los valores ajustados por el modelo "mod8". La línea roja horizontal indica valores de cero para los residuales.....	137
Figura 8. 13: residuales normalizados del modelo "mod8" en relación a los predictores: Localidad, Parcela, Fertilización con nitrógeno, Fertilización con azufre y días post antesis. De izquierda a derecha. La línea roja horizontal indica valores de cero para los residuales. ....	138
Figura 8. 14: gráfico de normalidad (cuantiles teóricos) para el modelo "mod8". ....	138
Figura 8. 15: residuales normalizados del modelo "Best" en relación a los ajustados por el modelo, los predictores: Localidad, Fertilización con nitrógeno, Fertilización con azufre, y días post antesis. Gráfico de normalidad (cuantiles teóricos) para el mismo modelo.....	143
Figura 8. 16: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Best". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.....	143
Figura 8. 17: residuales normalizados en relación a los valores ajustados por el modelo "Best". La línea roja horizontal indica valores de cero para los residuales. ....	144
Figura 8. 18: cantidad de hordeínas B en función de los días post antesis [log(dpa)]. los puntos indican los valores observados y las líneas, el ajuste general del modelo para los tratamientos con (N+. negro) y sin (N- azul) fertilización con nitrógeno.....	145

## Capítulo 9

Figura 9. 1: ubicación del sitio de estudio y fotos ilustrando la situación post-incendio y, durante la regeneración. ....	149
Figura 9. 2: distribución espacial de plántulas <i>Eupatorium buniifolium</i> .....	150
Figura 9. 3: distribución espacial de plántulas <i>Baccharis dracunculifolia</i> .....	150
Figura 9. 4: topografía de sitio de estudio .....	151
Figura 9. 5: distribución espacial de la densidad de plántulas <i>Baccharis dracunculifolia</i> .....	151
Figura 9. 6: distribución espacial de la densidad de plántulas <i>Baccharis dracunculifolia</i> .....	152
Figura 9. 7: distribución espacial de los residuales normalizados del modelo "den.gls". Los puntos rojos indican valores positivos y los negro indican valores negativos. El tamaño de los puntos indica la magnitud del residual. ....	153
Figura 9. 8: semivariograma de los residuales normalizados del modelo "den.gls".....	153
Figura 9. 9: semivariograma de los residuales normalizados del modelo "den.gls" en cuatro direcciones en el plano (0, 45, 90 y 135 °C).....	153
Figura 9. 10: semivariograma de los residuales normalizados del modelo "den.gls4" .....	155
Figura 9. 11: residuales normalizados (E4) del modelo "den.gls4" en relación a los ajustados (F4) por el modelo y los predictores: densidad de <i>Eupatorium buniifolium</i> muertas (densi\$Emuertas), de <i>Baccharis dracunculifolia</i> adultos (densi\$adulto) y la topografía (densi\$microtop). ....	156
Figura 9. 12: Gráfico de normalidad (cuantiles teóricos) para el mismo modelo "den.gls4.6" .....	158

## Capítulo 10

Figura 10. 1: esquema del diseño del experimento. El recuadro punteado rojo muestra la parte de los datos que utilizamos en este práctico.....	160
Figura 10. 2: residuales normalizados del modelo "mrM0" en relación a la profundidad.....	162
Figura 10. 3: semivariograma de los residuales normalizados del modelo " mrM0" .....	162

Figura 10. 4: residuales normalizados del modelo "mrM2.vE" en relación a los ajustados por el modelo, los predictores: profundidad, uso, parcela y bloque. Gráfico de normalidad (cuantiles teóricos) para el mismo modelo.....	166
Figura 10. 5: valores de nitrógeno del suelo ( $\text{g kg}^{-1}\text{suelo}$ ) para las distintas profundidades. Las barras muestran el valor medio $\pm$ error estándar.....	169

## Capítulo 11

Figura 11 1: relación entre variable cuantitativas: altitud, orientación Norte, orientación Este, pendiente, precipitación, temperatura media de verano, temperatura media de invierno, densidad de árboles de ciprés, edad promedio de los árboles de ciprés. Distribución de los puntos en el triángulo superior de la matriz y valor de coeficiente de Pearson en el triángulo inferior de la matriz. ....	173
Figura 11 2: residuales en relación a los valores ajustados por el modelo "modelGlobA" .....	175
Figura 11 3: residuales del modelo "modelGlobA" en relación a las variables predictoras: región, orientación Norte, pendiente, altitud, temperatura media de verano y temperatura media de invierno. ....	176
Figura 11 4: residuales del modelo "modelGlobA" en relación a las variables predictoras: precipitación, densidad y edad promedio de árboles de ciprés. Gráfico de normalidad en la fila inferior.....	176
Figura 11 5: distribución espacial de los residuales normalizados del modelo "modelGlobA". Los puntos rojos indican valores positivos y los negro indican valores negativos. El tamaño de los puntos indica la magnitud del residual. ....	177
Figura 11 6: semivariograma de los residuales normalizados del modelo " GlobAlme" .....	178
Figura 11 7: residuales estandarizados en relación a los valores ajustados por los modelos "GlobAlme" (inicial. Izq.), " modelVp6" (mejor modelo con función de la varianza. Centro) y, "modFinal" (mejor modelo según inferencia multimodelo. Der.).....	186
Figura 11 8: residuales del modelo "modFinal" en relación a las variables predictoras: Región, edad promedio de los árboles de ciprés, temperatura media de verano y temperatura media de invierno.....	187
Figura 11 9: Gráfico de normalidad (cuantiles teóricos) para el mismo modelo "modFinal".	187
Figura 11 10: distribución espacial de los residuales normalizados del modelo "modFinal". Los puntos rojos indican valores positivos y los negro indican valores negativos. El tamaño de los puntos indica la magnitud del residual. ....	188
Figura 11 11: valores observados de volumen de ciprés ( $\text{m}^3/\text{árbol}$ ) en función de los valores ajustados por el modelo "modFinal" según el método de inferencia multimodelo. ....	189
Figura 11 12: volumen de ciprés ( $\text{m}^3$ ) en función de la Edad promedio (Izq.) de la temperatura media de verano (Centro) y la temperatura media de invierno (Der.). Los puntos grises muestran los valores observados y los negros, los predichos por el modelo "modFinal". ....	191
Figura 11 13: volumen de ciprés ( $\text{m}^3$ ) en función de la temperatura media de invierno ( $^{\circ}\text{C}$ ) y la edad promedio (años). Los puntos grises muestran los valores observados y los verdes, los predichos por el modelo "modFinal" .....	191

## Capítulo 12

Figura 12. 1: diseño del experimento.....	194
Figura 12. 2: cantidad de capítulos en relación a la altura de las plantas de <i>Baccharis dracunculifolia</i> . Los colores indican distintos sitios de muestreo identificados con el año de ocurrencia del último incendio: 1965 (negro), 19779 (rojo), 1997 (verde), 2000 (azul).....	196

Figura 12. 3: probabilidad de presencia (presencia/ ausencia) de capítulos en relación a la altura de las plantas de <i>Baccharis dracunculifolia</i> . Los colores indican distintos sitios de muestreo identificados con el año de ocurrencia del último incendio: 1965 (negro), 19779 (rojo), 1997 (verde), 2000 (azul).....	196
Figura 12. 4: efecto aleatorio (diferencia respecto de la intercepción) estimado por el modelo "M.0" para cada uno de los sitios (1965, 1979, 1997, 2000). .....	203
Figura 12. 5: probabilidad de presencia de capítulos en función de la altura de plantas de <i>Baccharis dracunculifolia</i> estimada por el modelo "M.0". La línea llena muestra la probabilidad de encontrar un arbusto de baccharis con capítulos en cualquiera de los sitios estimado por el modelo y las líneas punteadas el intervalo de confianza de la estimación.....	204
Figura 12. 6: probabilidad de presencia de capítulos en función de la altura de plantas de <i>Baccharis dracunculifolia</i> estimada por el modelo "M.0". La línea llena muestra la probabilidad de encontrar un arbusto de baccharis con capítulos en cualquiera de los sitios estimado por el modelo y las líneas de colores, la probabilidad para cada uno de los sitios según la intercepción aleatoria estimada.....	206
<b>Capítulo 13</b>	
Figura 13. 1: diseño del experimento.....	210
Figura 13. 2: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "ModPQL". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cuál se considera que hay autocorrelación entre las mediciones.....	217
Figura 13. 3: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Mod". Las líneas llenas verticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.....	218
Figura 13. 4: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "ModPQL". Las líneas llenas veticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.....	218
Figura 13. 5: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "ModADMB". Las líneas llenas veticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.....	219
Figura 13. 6: gráfico de autocorrelación en relación a las distancias entre mediciones (lag) para el modelo "Mod2". Las líneas llenas veticales indican el nivel de significancia y las líneas punteada el umbral por sobre el cual se considera que hay autocorrelación entre las mediciones.....	219
Figura 13. 7: residuales normalizados (o "pearson") en relación a los valores ajustados por los modelos glmer (Mod, family="poisson"), glmmPQL (ModPQL, family= "poisson"), glmmADMB (ModADMB_nb, family="nbinom") y glmer [Mod2, family="poisson" con corrección para sobredispersión según Bolker et al. (2009) y Gelman y Hill (2007)]. La línea roja indica residual cero. Observar que la escala en el eje de las ordenadas no es idéntico entre gráficos. ....	220
Figura 13. 8: valores observados de número de granos en relación a los valores ajustados por los modelos glmer (Mod, family="poisson"), glmmPQL (ModPQL, family= "poisson"), glmmADMB (ModADMB_nb, family="nbinom") y glmer [Mod2, family="poisson" con corrección para sobredispersión según Bolker et al. (2009) y Gelman y Hill (2007)]. La línea roja indica la relación 1:1.....	222

**Capítulo 14**

Figura 14. 1: diseño del experimento.....	227
---	-----

Datos jerárquicos



## Anexo

### Requerimientos previos

Pueden venir al curso con sus computadoras portátiles.

En ellas deberán tener instalados los programas R (<http://www.r-project.org/>) y R Studio (<http://www.rstudio.com/>). Además, les pedimos que instalen una serie de “paquetes” que contienen las funciones que utilizaremos a los largo del curso.

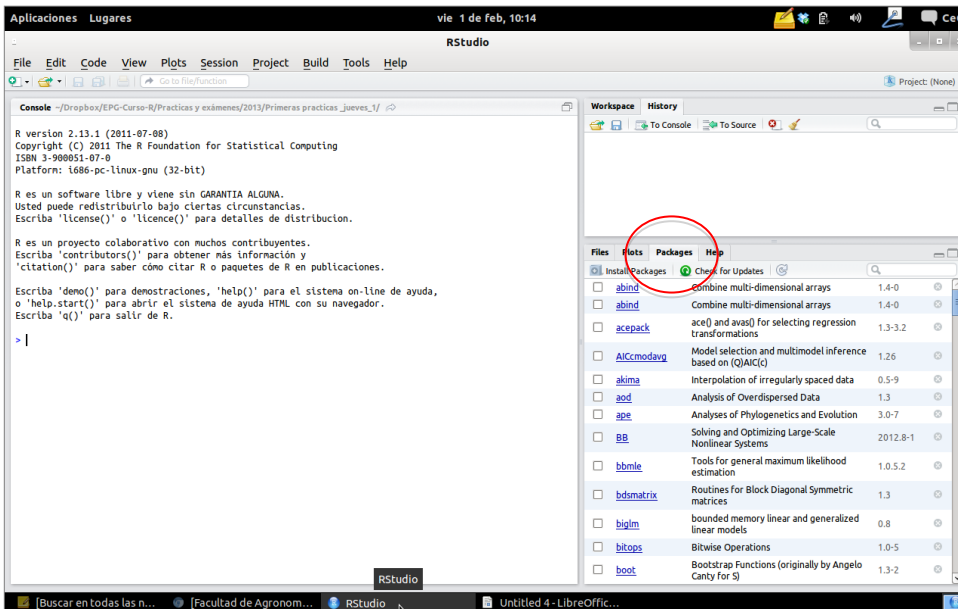
Los paquetes que deberán tener instalados son los siguientes:

- aods3 (solo para evaluar dispersión en Capítulo 11)
- bbmle
- boot (solo para Capítulo 10)
- car
- coefplot2
- glmmADMB
- gstat
- lattice
- lme4
- MASS
- multcomp
- MuMin (elegir mirror Brasil (PR) no todos funcionan con modelos mixtos)
- nlme
- rgl
- RLRsim
- sp

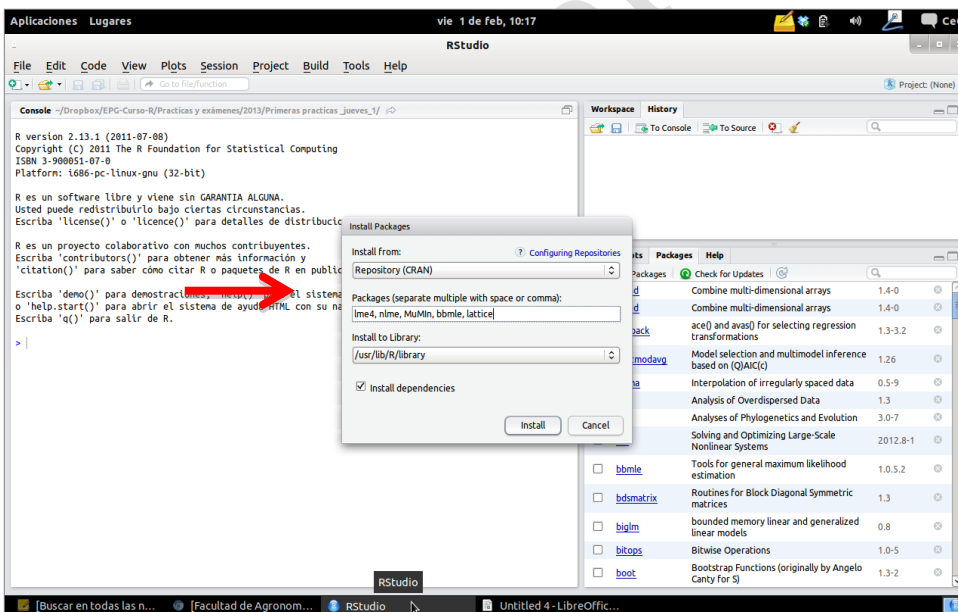
## ¿Cómo instalar paquetes desde RStudio?

Si bien existen diferentes maneras de instalar paquetes en R, describimos a continuación una de las maneras que nos parece más sencilla. Si esta manera no les resulta, no duden en consultarnos.

Al abrir el programa RStudio verán una solapa en el cuadrante derecho inferior llamada “Packages”.



Dentro de esta solapa verán el botón “Install Packages”



En la segunda opción “Packages” pueden escribir los paquetes necesarios separados por comas. Es importantes que este seleccionada la opción “Install dependencies” al pie de la ventana. Finalmente, presionar el botón “Install”.

El repositorio es el sitio web desde donde se descargan los archivos. R posee una lista extensa de repositorios a lo largo de todo el mundo. Si tienen que, o desean modificar el repositorio en

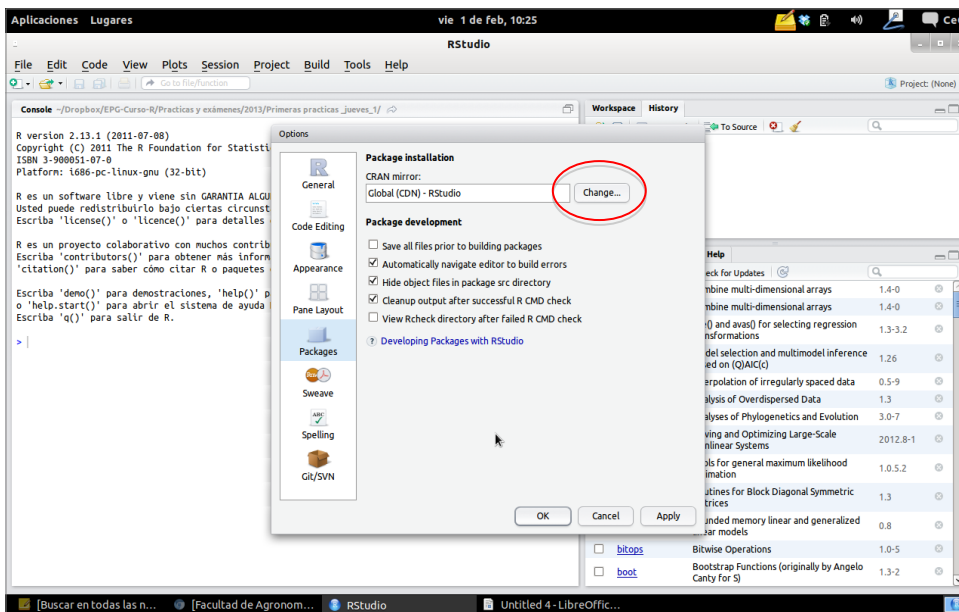


“Tools”>”Options”> “Packages” encontrarán la opción “CRAN mirror” (i.e.: repositorio), Change....

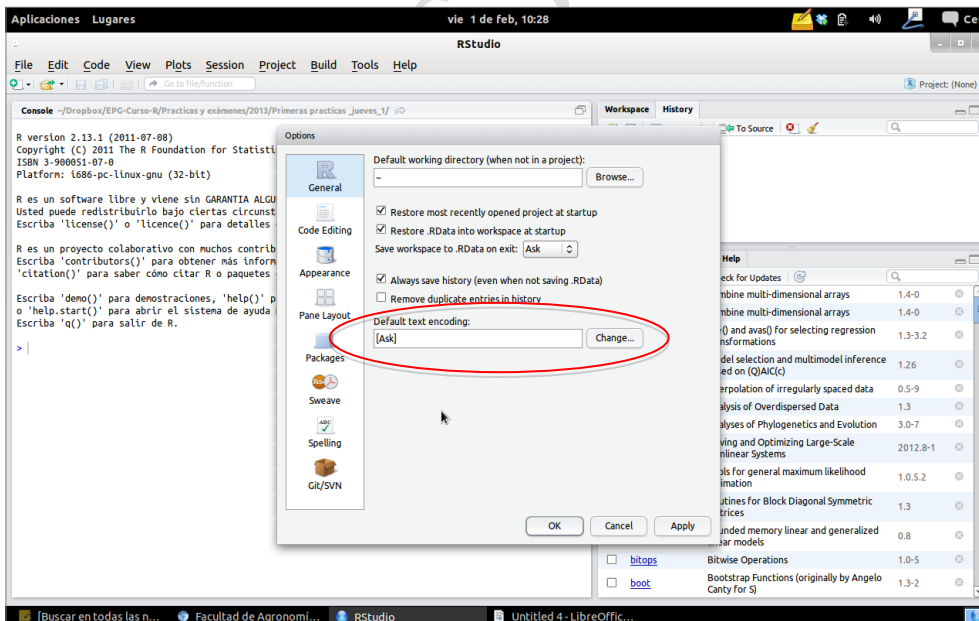
Si utilizan el sistema operativo **Ubuntu**, pueden instalar las paquetes desde el “Centro de software”. En búsqueda deben poner “cran” y el paquete que quieran instalar (Ej.: cran lme).

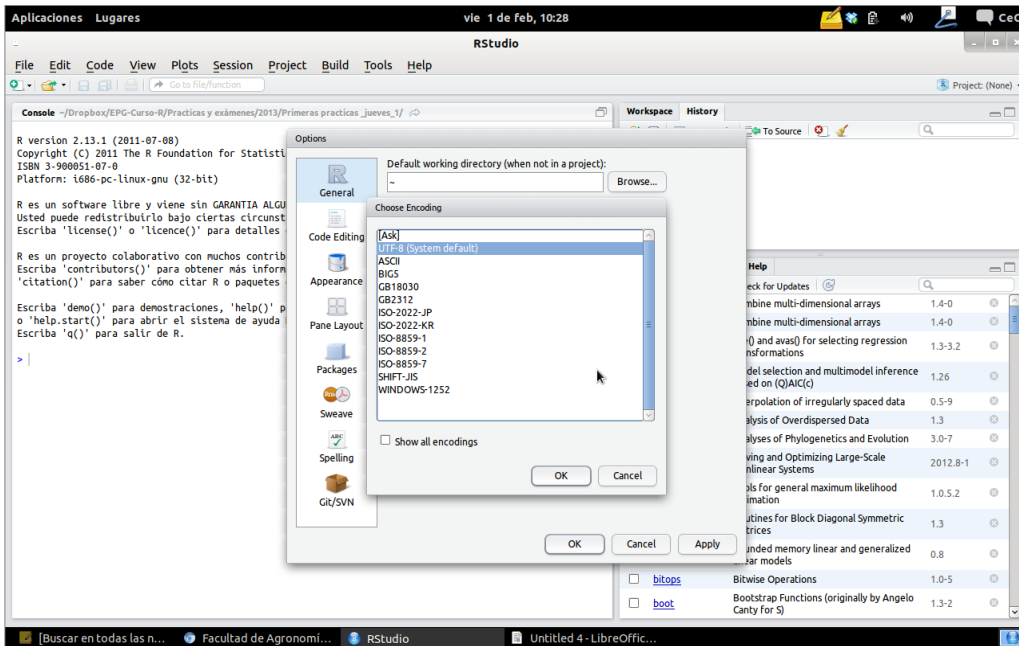
### ¿Cómo modificar la codificación de las sentencias?

Si tienen que cambiar la codificación en la misma sección “Tools”>”Options”, encontrarán > “General” > “Default text encoding”, Change....



**Recuerden seleccionar UTF-8.**





La estética de las ventanas puede cambiar levemente dependiendo del sistema operativo utilizado.