Tutorial 11: Anova unifactorial

Atención:

- Este documento pdf lleva adjuntos algunos de los ficheros de datos necesarios. Y está pensado para trabajar con él directamente en tu ordenador. Al usarlo en la pantalla, si es necesario, puedes aumentar alguna de las figuras para ver los detalles. Antes de imprimirlo, piensa si es necesario. Los árboles y nosotros te lo agradeceremos.
- Fecha: 10 de septiembre de 2015. Si este fichero tiene más de un año, puede resultar obsoleto. Busca si existe una versión más reciente.

Índice

1. Los ingredientes básicos del Anova unifactorial en R

En el Ejemplo 11.1.1 (pág. 417) del Capítulo 11 del libro, teníamos cuatro muestras de 100 frailecillos, a cada una de las cuales aplicábamos un tratamiento distinto y medíamos las respuestas en aleteos por minuto. Los resultados se recogían en la que aquí incluimos como Tabla 1 (sólo se muestran las primeras seis filas, la tabla completa tiene un total de 100 filas):

	Aliron	Vuelagra	Plumiprofeno	Elevantolin
1	76.65	76.74	87.14	88.66
2	79.36	74.72	82.34	78.12
3	71.83	68.61	94.06	81.74
4	73.24	72.84	88.12	89.11
5	79.73	75.83	84.47	82.90
6	74.50	66.81	83.11	80.84

Tabla 1: Tabla **defectuosa** del Ejemplo 11.1.1 del libro.

Y allí dijimos que, por visualmente atractivo que este formato pueda resultar, no es el más adecuado para almacenar los datos en el ordenador, y operar con ellos. En este contexto se usa a veces la terminología de datos sucios / datos limpios (en inglés *messy data* / *tidy data*). Si el lector está interesado, el artículo de Hadley Wickham que puedes descargar desde:

http://vita.had.co.nz/papers/tidy-data.pdf

es una referencia para adentrarse en ese terreno (aunque no es necesario leerlo para este tutorial). En los llamados datos limpios, además de otras condiciones que aquí no vamos a usar, se exige que:

1. Cada variable aparezca en una columna de la tabla.
2. Cada observación individual aparezca en una fila de la tabla.

Y, como puedes ver, eso no sucede en la Tabla 1. Los *títulos* de las columnas se corresponden con los niveles (valores) del factor `Tratamiento`. En cambio, la Tabla 2 contiene esos mismos datos en un formato *limpio*. Ahora la primera columna corresponde al nivel del tratamiento, y la segunda

	tratamiento	respuesta
1	Aliron	76.65
2	Elevantolin	88.66
3	Aliron	79.36
4	Vuelagra	76.74
5	Aliron	71.83
6	Vuelagra	74.72
7	Plumiprofeno	87.14
8	Aliron	73.24
9	Elevantolin	78.12
10	Plumiprofeno	82.34

Tabla 2: Tabla bien construida (limpia) para el Ejemplo 11.1.1.

a la respuesta. Y en cada fila de la tabla nos referimos a un frailecillo individual, y anotamos el tratamiento al que fue sometido, y cual fue su respuesta (en aleteos/minuto). El fichero

Cap11-frailecillos.csv

contiene los datos limpios de ese ejemplo, y vamos a usarlo para ilustrar la forma de realizar el contraste Anova con R. En la Sección 4 de este tutorial (pág. 16) volveremos sobre este tema, para plantearnos una pregunta de bastante importancia práctica: si los datos de partida de los que disponemos no son limpios, en este sentido, ¿cómo podemos limpiarlos?

1.1. Preliminares: lectura de datos.

Volviendo al caso en que partimos de un fichero "limpio" como `Cap11-frailecillos.csv`, en esta sección vamos a describir el fichero

Tut11-Anova-Basico.R

El objetivo de este fichero es calcular el contraste Anova para un caso sencillo como el del Ejemplo 11.1.1 del libro. La única salvedad que vamos a hacer es que, en el fichero, vamos a permitir que el número de observaciones de cada nivel del factor sea distinto (en el Ejemplo 11.1.1 eran todos del mismo tamaño, cuatro niveles, cada uno con 100 observaciones).

En el resto de esta sección vamos a comentar el funcionamiento de ese fichero, deteniéndonos más en las partes más relevantes. La primera parte del fichero es, simplemente, la lectura de los datos, que suponemos almacenados en un fichero csv *limpio*, en el sentido que hemos descrito. Para usar el fichero correctamente asegúrate de leer las instrucciones que parecen en los comentarios iniciales.

El contenido del fichero `csv` se almacenará en un `data.frame` llamado `datos.` Además, en este primer bloque se incluye, comentada por si deseas usarla, una llamada a la función `View` para verificar la lectura correcta de los datos.

En nuestro caso las líneas de lectura del fichero quedarían así:

```
########################################
# LECTURA DE LOS DATOS
########################################

#Leemos los datos del fichero
datos = read.table(file="./datos/Cap11-frailecillos.csv", sep=" ",header=T)
```

```r
# Descomenta esta línea si quieres comprobar que la lectura ha sido correcta.
# View(datos)

# IMPORTANTE: introduce el número de columna ( 1 o 2 ) que contiene el factor (tratamiento).
colFactor = 1
```

Una vez concluida la lectura de los datos, comenzamos a analizarlos. Hay que tener en cuenta que, en cada ejemplo concreto, los nombres de las variables (nombres de las columnas del fichero `csv`) serán distintos. Para facilitar el trabajo, lo primero que hace el código de este fichero es renombrar las columnas para que siempre se llamen `Tratamiento` (la que contiene el factor) y `Respuesta` (la que contiene, obviamente, la variable cuantitativa de respuesta). Para renombrarlas usaremos la función `colnames`, pero además debemos tener en cuenta en qué columna se halla el factor y en cuál la respuesta. Por eso, el renombrado se hace a través de una estructura `if/else`. El resultado es que el `data.frame datos` ahora tiene dos variables, a las que nos podemos referir como:

- `datos$Tratamiento`, que es el factor.
- `datos$Respuesta`, la variable cuantitativa.

A continuación, guardamos en la variable `niveles` los *nombres* o *etiquetas* de los niveles del factor `Tratamiento`, usando para ello la función `levels`. Además, vamos a almacenar en la variable `N` el número total de observaciones (que coincide con el número de filas del `data.frame`, al tratarse de datos *limpios*), obtenido con `nrow`, mientras que, en la variable `k`, almacenaremos el número del niveles del factor. En el caso del Ejemplo 11.1.1 sería así:

```r
##########################################
# ANALISIS DE LOS DATOS
# No es necesario cambiar nada de aquí para abajo.
##########################################

# Renombramos las columnas del data.frame
if(colFactor==1){
  colnames(datos) = c("Tratamiento","Respuesta")
}else{
  colnames(datos) = c("Respuesta","Tratamiento")
}
colnames(datos)

## [1] "Tratamiento" "Respuesta"

# Etiquetas de los niveles del factor.

(niveles=levels(datos$Tratamiento))

## [1] "Aliron"       "Elevantolin"  "Plumiprofeno" "Vuelagra"

# El número total de observaciones.

(N=nrow(datos))

## [1] 400

# Calculamos el número de niveles del factor

(k=length(niveles))

## [1] 4
```

1.2. Construcción manual de la tabla Anova

En el trabajo con R, lo habitual es usar las funciones `lm` y `anova`, que veremos después, para obtener una Tabla Anova (ver la Tabla 11.3 del libro, pág. 428). Pero en un primer contacto con los contrastes Anova es muy importante entender el contenido de esa tabla. Y no hay mejor manera de entenderlo, a mi juicio, que programar nosotros mismos la construcción de la tabla. Por esa razón, a continuación vamos a reconstruir, paso a paso, los elementos que aparecen en la Identidad Anova (Ecuación 11.3, pág. 425). Por eso, hasta nuevo aviso, la salida que se muestra corresponde a los datos del Ejemplo 11.1.1 del libro. Ten en cuenta que, por los detalles técnicos de la forma en la que se incluyen los cálculos en este Tutorial, el redondeo puede ser distinto aquí del que se muestra en el libro.

El primer paso es construir SST (también llamado $SStotal$), que es:

$$SST = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (X_{ij} - \bar{X})^2.$$

Para eso, necesitamos $\bar{X}$, que es la media de la variable respuesta. Vamos a ver el código correspondiente en R:

```
########################################
# Construccion manual de la tabla ANOVA

# La media de todas las respuestas, sin distinguir niveles del factor

(mediaDatos = mean(datos$Respuesta))

## [1] 78.825

# La suma total de cuadrados.

(SStotal = sum((datos$Respuesta-mediaDatos)^2) )

## [1] 14881
```

En ese primer paso no hay grandes novedades. El segundo paso es más interesante. Vamos a calcular $SSmodelo$, que es:

$$SSmodelo = \sum_{j=1}^{k} n_j \cdot (\bar{X}_{\cdot j} - \bar{X})^2.$$

Para esto, vamos a tener que calcular la media de la variable respuesta para cada nivel del tratamiento. Afortunadamente, la función `tapply`, que vimos en la Sección ?? del Tutorial08 nos permite hacer esto fácilmente.

```
# La media de cada uno de los niveles

(mediasPorNivel = tapply(datos$Respuesta, datos$Tratamiento, mean))

##       Aliron  Elevantolin Plumiprofeno     Vuelagra
##       78.399       80.400       84.400       72.100
```

Además necesitamos calcular los valores n_j, que son el número de réplicas. Es decir, cuántas observaciones tenemos para cada nivel del factor. En general, no tiene porque suceder que todos los n_j sean iguales (recuerda que, cuando sucede, se dice que el diseño es equilibrado). Esto tampoco es demasiado complicado de conseguir: el número de veces que se repite un nivel es su frecuencia, así que basta con obtener una tabla de frecuencias:

```
# Cuantos elementos hay en cada nivel

(replicas = table(datos$Tratamiento) )
```

```
##
##       Aliron  Elevantolin Plumiprofeno     Vuelagra
##          100          100          100          100
```

Y, con esto, el cálculo de `SSmodelo` es muy sencillo:

```
# Ya podemos  calcular la suma de cuadrados del modelos

(SSmodelo = sum( replicas * (mediasPorNivel-mediaDatos)^2  ) )

## [1] 7897
```

Un detalle técnico: `replicas` es un objeto de clase `table`. Puedes comprobarlo usando

```
class(replicas)

## [1] "table"
```

Pero en el cálculo de *SSmodelo* lo hemos usado como si fuera un vector, y R se ha encargado de hacer los ajustes necesarios para que todo funcione correctamente.

Hemos dejado para el final el paso más interesante, el cálculo de *SSresidual*, porque incluye un detalle novedoso (para nosotros) sobre el manejo de factores. Empecemos recordando que:

$$SSresidual = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (X_{ij} - \bar{X}_{\cdot j})^2.$$

Y la dificultad para calcular esto, a partir de la estructura de datos que tenemos en el `data.frame datos` es que, para cada fila de `datos` tenemos que localizar la media $\bar{X}_{\cdot j}$ del grupo (nivel) al que corresponde esa observación. Conviene recordar que esas medias ya las hemos calculado, y están almacenadas en el vector `mediasPorNivel`. Para entender lo que tenemos que hacer, fíjate en una fila cualquiera del `data.frame datos`, por ejemplo la fila 7:

```
datos[7,]

##    Tratamiento Respuesta
## 7 Plumiprofeno     87.14
```

En este caso, el nivel `Plumiprofeno` es, en la ordenación que R ha hecho, el tercer nivel del factor `datos$Tratamiento`. Así que, al valor `87.14` de la respuesta tenemos que restarle el tercer elemento de `mediasPorNivel`. Pero ¿cómo convertimos el valor `Plumiprofeno` (que es de tipo `character`) en el número 3 que necesitamos para poder decirle a R que use el tercer elemento? Pues usando la función `as.numeric`, que nos devuelve la representación numérica de los niveles del factor, en la ordenación que R está usando. Concretando, vamos a almacenar esa representación numérica de los niveles en un vactor llamado `numeroNivel`:

```
# Para calcular la suma residual vamos a crear un vector que nos dice,
# para cada fila, cual es el número de nivel del factor que le corresponde.

numeroNivel = as.numeric(datos$Tratamiento)
```

```
head(numeroNivel,10)

##  [1] 1 2 1 4 1 4 3 1 2 3
```

Aunque no es necesario para el funcionamiento del programa, a efectos de ilustración hemos incluido (aquí, en el Tutorial, pero no en el fichero) la función `head` para mostrar los primeros diez elementos

de `numeroNivel`. Como ves, ese vector hace exactamente lo que queríamos, y nos dice, para cada fila de `datos`, cuál es el número del correspondiente nivel de `datos$Tratamiento`.

Una vez hecho esto, el cálculo de la suma residual es fácil:

```
# Y lo usamos para calcular la suma de cuadrados residual

(SSresidual=sum( ( datos$Respuesta -  mediasPorNivel[numeroNivel] )^2 ) )

## [1] 6984.4
```

Y la constatación de la identidad Anova es la igualdad entre estos dos resultados:

```
# La identidad ANOVA es la coincidencia entre los dos siguientes numeros:

SStotal

## [1] 14881

SSmodelo + SSresidual

## [1] 14881
```

El siguiente paso, en la construcción de la Tabla Anova, es el cálculo del estadístico Ξ de la Ecuación 11.4 (pág. 427 del libro).

$$\Xi = \frac{\frac{SS_{\text{modelo}}}{k-1}}{\frac{SS_{\text{residual}}}{N-k}}.$$

Así que hacemos:

```
# Ahora podemos calcular el estadistico

(Estadistico = ( SSmodelo / (k - 1) ) / ( SSresidual / ( N - k ) ))

## [1] 149.25
```

Y, para terminar, podemos calcular el p-valor usando el hecho de que Ξ se comporta como una variable de tipo F de Fisher, concretamente:

$$\Xi \sim F_{k-1;N-k}.$$

Así que el p-valor se calcula con:

```
# Y el p-valor:

(pValor=1-pf(Estadistico,df1= k-1, df2= N-k ))

## [1] 0
```

La respuesta igual a 0 que obtenemos significa que el p-valor, para el Ejemplo 11.1.1 en concreto, es tan pequeño que R lo identifica con 0. La conclusión, evidentemente, es que el contraste es significativo, rechazamos la hipótesis nula, y podemos decir que, basándonos en estos datos, las respuestas medias de cada uno de los niveles no son todas iguales. Recuerda que esto no significa que sean todas distintas.

Ejercicio 1.

1. *El Ejemplo 11.6.4 del libro (pág. 447) incluye el fichero adjunto*

   Cap11-ComparacionesPostHoc.csv

que contiene una tabla de datos (son datos limpios*, en el sentido que se discute en la Sección 4, pág. 16), con $N = 140$ valores de una variable continua llamada* `respuesta`*, correspondientes a seis niveles diferentes de un factor llamado* `tratamiento` *(los seis niveles se llaman* `grupo1, grupo2,` *etc.) Usa los datos de ese fichero con las instrucciones R del fichero*

Tut11-Anova-Basico.R

que hemos visto en este tutorial para, en primer lugar, construir "manualmente" la tabla Anova correspondiente a ese ejemplo. En las próximas secciones de este tutorial vamos a analizar el resto de las instrucciones del fichero (desde el bloque de Representación gráfica *en adelante), así que no te preocupes si todavía no entiendes alguna de las instrucciones en los últimos bloques de este fichero R.*

2. *Haz lo mismo con el fichero*

Cap11-AnovaSignificativoPostHocNoSignificativo.csv

que aparece en el Ejemplo 11.6.5 del libro (pág. 448).

□

Gráficos boxplot por niveles del tratamiento.

A continuación vamos a representar gráficamente los datos de este Ejemplo, recordando algo que ya vimos en el Tutorial07: la posibilidad de usar el factor `datos$Tratamiento` para obtener diagramas de caja (boxplots) paralelos de las respuestas para cada uno de los niveles del tratamiento. Hacemos:

```
##############################
# Representacion grafica.
##############################

# Boxplots paralelos de los datos
par(font.axis=2, cex.axis=1.5, lwd=2)
boxplot(datos$Respuesta ~ datos$Tratamiento, col=terrain.colors(k), notch = TRUE)
```

y obtenemos el gráfico que se muestra en la Figura 1 (en el que se muestra aquí hemos modificado algunos parámetros gráficos, para lograr una mejor visualización).

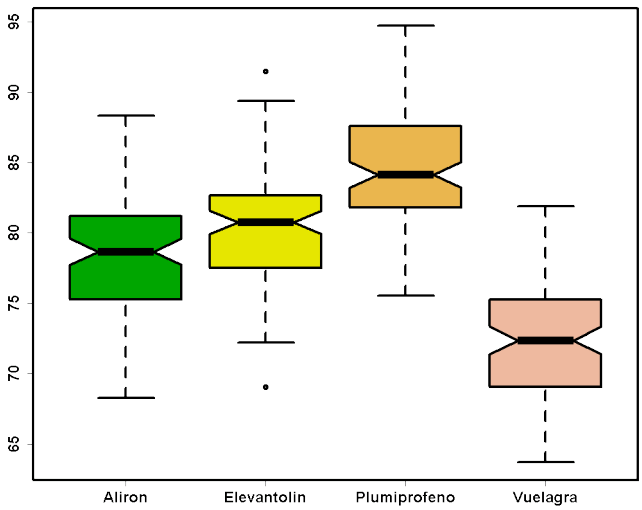

Figura 1: Boxplots paralelos del Ejemplo 11.1.1 del libro.

Como puedes ver, estos diagramas de caja tienen una *muesca* o *bisel doble* alrededor de la mediana. La anchura de esa muesca viene a ser un análogo de los intervalos de confianza que conocemos, pero aplicado a la mediana en lugar de la media. El parámetro `notch` controla la aparición de ese bisel doble o muesca en el diagrama de cajas. ¿Por qué lo hemos incluido en este caso? Pues porque, al tratarse de muestras grandes y, como veremos en la próxima sección, aproximadamente normales, la media y la mediana son muy similares. Y esos biseles permiten, de forma muy sencilla, evaluar gráficamente si existen diferencias significativas entre las medianas (y por consiguiente, en este ejemplo, entre las medias). La regla práctica que se aplica es que si los biseles correspondientes a dos niveles no se solapan entre sí (en vertical), entonces podemos sospechar que existen diferencias significativas entre las medianas correspondientes a esos dos niveles. Si quieres profundizar más en el uso y significado del parámetro `notch` te recomendamos que consultes la ayuda de la función `boxplot` de R.

Como puede verse, en este ejemplo tenemos razones para sospechar que existen diferencias significativas, dos a dos, en todos los casos: no hay dos niveles que tengan la misma media (aunque *Alirón* y *Elevantolín* tienen medias bastante similares). El gráfico también parece responder a la pregunta de cuál es el tratamiento más eficaz, ya que *Plumiprofeno* parece ser significativamente mejor que los demás. Volveremos sobre esa discusión en la Sección 3 (pág. 13).

Ejercicio 2. *Obten las figuras correspondientes para los ficheros de datos que hemos usado en el Ejercicio 1, pág. 6.* □

Coeficiente de correlación lineal.

Para completar el cálculo manual de la tabla Anova vamos a obtener el valor del coeficiente de correlación lineal y el valor del coeficiente de correlación ajustado que se describen en la Sección 11.4.1 del libro (pág. 436). El fichero `Tut11-Anova-Basico.R` contiene asimismo instrucciones para calcular "a mano" esos coeficientes a partir de nuestros cálculos previos del modelo:

```
###############################################
# Coeficiente de correlacion.
###############################################

# El coeficiente de correlacion lineal es:

(R2 = SSmodelo / SStotal)

## [1] 0.53066

# Mientras que el coeficiente de correlacion lineal ajustado es:

(adjR2 = 1 - ((SSresidual / (N - k)) / (SStotal / (N - 1))))

## [1] 0.52711
```

Ejercicio 3. *Obten la tabla Anova y el coeficiente de correlación lineal para cada uno de los ficheros de datos que hemos usado en el Ejercicio 1, pág. 6.* □

1.3. Tabla Anova con las funciones `lm` y `anova` de R. Coeficiente de correlación y coeficientes del modelo lineal.

Aunque en la sección anterior hemos obtenido uno por uno los elementos que componen la Tabla de un contraste Anova unifactorial, esa no es, desde luego, la forma habitual de proceder. En la práctica resulta mucho más cómodo recurrir a las funciones que proporciona R y que en apenas dos líneas de código reproducen todos esos pasos que, laboriosamente, hemos ido dando.

Concretamente, el primer paso es utilizar la función `lm`, que tuvimos ocasión de conocer en el Tutorial10, y después aplicaremos la función `anova` al modelo resultante de `lm`. Hacemos:

```
#################################################
# Ahora vamos a obtener los mismos resultados
# usando las funciones lm,  anova y summary:
#################################################

datos.lm = lm(Respuesta ~ Tratamiento, data = datos)

anova(datos.lm)

## Analysis of Variance Table
##
## Response: Respuesta
##              Df Sum Sq Mean Sq F value Pr(>F)
## Tratamiento   3   7897    2632     149 <2e-16 ***
## Residuals   396   6984      18
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Una ventaja añadida de nuestro trabajo de la sección anterior es que ahora resulta muy fácil interpretar esta Tabla Anova, y ver cuál es la información que R produce como respuesta.

- Las dos primeras columnas contienen, respectivamente, los grados de libertad `Df`, y las sumas de cuadrados del modelo y residual, en la que se denomina `Sum Sq`.
- La única columna de la tabla que puede despistarnos un poco es la tercera, que tiene como encabezamiento `Mean Sq`, pero en ella aparecen, simplemente, el numerador y el denominador, respectivamente, del Estadístico Ξ, que a su vez aparece en la columna denominada `F value`.
- El p-valor aparece en la columna `Pr(>F)` (el símbolo, como ves, indica la cola derecha de la distribución F). R utiliza, como en otros casos, un código de asteriscos para indicar el tamaño del p-valor que se ha obtenido. En este caso, el valor `2.2e-16` que aparece no es, realmente, el p-valor, sino una forma de indicarnos que se ha alcanzado un valor más pequeño que el límite de precisión de R para distinguirlo del cero (y que, desde luego, justifica sobradamente los tres asteriscos).

Otra forma de obtener información útil adicional sobre el modelo es usando la función `summary`.

```
summary(datos.lm)

##
## Call:
## lm(formula = Respuesta ~ Tratamiento, data = datos)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11.37  -2.81   0.11   2.77  11.11
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                78.399      0.420  186.68  < 2e-16 ***
## TratamientoElevantolin      2.000      0.594    3.37  0.00083 ***
## TratamientoPlumiprofeno     6.001      0.594   10.10  < 2e-16 ***
## TratamientoVuelagra        -6.299      0.594  -10.61  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.2 on 396 degrees of freedom
## Multiple R-squared:  0.531,Adjusted R-squared:  0.527
## F-statistic:  149 on 3 and 396 DF,  p-value: <2e-16
```

Volviendo a los resultados de la función `summary`, cabe preguntarse qué representan los números que aparecen en el apartado `Coefficients`. Pero para hacerlo tendremos que esperar un poco, porque necesitamos las ideas sobre *constrastes* que se discuten en la Sección 11.6.2 del libro (pág. 453) y en la Sección 5 de este tutorial (pág. 18). En cualquier caso, para adelantar un poco esas ideas, puedes observar que el primer valor que aparece en la fila `Intercept` es la media $\bar{X}_{\cdot 1} = 78.40$ del primer nivel (por orden alfabético), el *Aliron*. Los valores iniciales de las siguientes tres filas son *las diferencias* entre este valor y las medias de los restantes tres niveles:

$$\begin{cases} \bar{X}_{\cdot 2} - \bar{X}_{\cdot 1} = 80.40 - 78.40 = 2.00 \\ \bar{X}_{\cdot 3} - \bar{X}_{\cdot 1} = 84.40 - 78.40 = 6.00 \\ \bar{X}_{\cdot 4} - \bar{X}_{\cdot 1} = 72.10 - 78.40 = -6.30 \end{cases}$$

Más adelante, en la Sección 5 de este Tutorial, veremos por qué nos pueden interesar estos valores, y qué significan las restantes columnas de la tabla en la que aparecen. En particular, aprenderemos a controlar el proceso. Al construir esta tabla R ha tomado muchas decisiones de manera automática. Por ejemplo, ha decidido comparar $\bar{X}_{\cdot 1}$ con los demás. Esas decisiones pueden no coincidir con nuestros deseos a la hora de analizar un diseño experimental concreto. A lo mejor a nosotros nos interesa más comparar $\bar{X}_{\cdot 2}$ frente a los restantes valores, o queremos averiguar si las medias están ordenadas de determinada manera, etc.

2. Verificando las condiciones del Anova unifactorial en R

Puesto que la discusión de las condiciones de aplicabilidad del contraste Anova que hemos hecho en la Sección 11.5 (pág. 437) del libro ha sido bastante superficial, no vamos a entrar aquí en un análisis muy minucioso de cómo usar R para chequear esas condiciones. Nos vamos a limitar a las ideas más básicas. Para facilitar el trabajo, vamos a incluir el código necesario en un fichero, llamado

Tut11-Anova-Avanzado.R

Este fichero contiene parte del mismo código que hemos visto en el fichero `Tut11-Anova-Basico.R`, del que hemos eliminado la parte de cálculo paso a paso de la Tabla Anova, dejando sólo la versión que usa las funciones `lm` y `anova`. Y hemos añadido código para un análisis más detallado del contraste Anova, que vamos a describir en ésta y en la próxima sección.

Histogramas por niveles (grupos).

Hemos visto ya como obtener diagramas de cajas paralelos de la respuesta para los distintos grupos (niveles) del factor `Tratamiento`. Esos diagramas de cajas, si los grupos son suficientemente numerosos, pueden servir para verificar gráficamente la condición de normalidad. Además, la dispersión dentro de cada uno de los grupos no debería ser llamativamente distinta, para que se cumpla la condición de homogeneidad de varianzas. Se puede hacer un estudio similar a partir de los histogramas de la respuesta en cada uno de los grupos. Para obtener todos los histogramas, de nuevo vamos a solicitar la ayuda de `tapply`. El código, del que enseguida comentaremos algunos detalles, es este:

```
# Histogramas de cada uno de los grupos
par(mfrow=c(k%/%2+k%%2,2),font.axis=1.5,cex.axis=1.5,lwd=2,font.lab=2,cex.lab=1.5)
histogramas = tapply(datos$Respuesta,datos$Tratamiento, hist, breaks=10,
       main="", ylab="Frecuencia", xlab="", col=heat.colors(10))
par(mfrow=c(1,1))
```

y el resultado puede verse en la Figura 2.

Algunos comentarios sobre esa figura y sobre el código que hemos usado para generarla:

- Los cuatro histogramas que hemos obtenido, uno por nivel, no parecen indicar que haya ningún problema evidente con las condiciones de normalidad y homogeneidad de las varianzas.

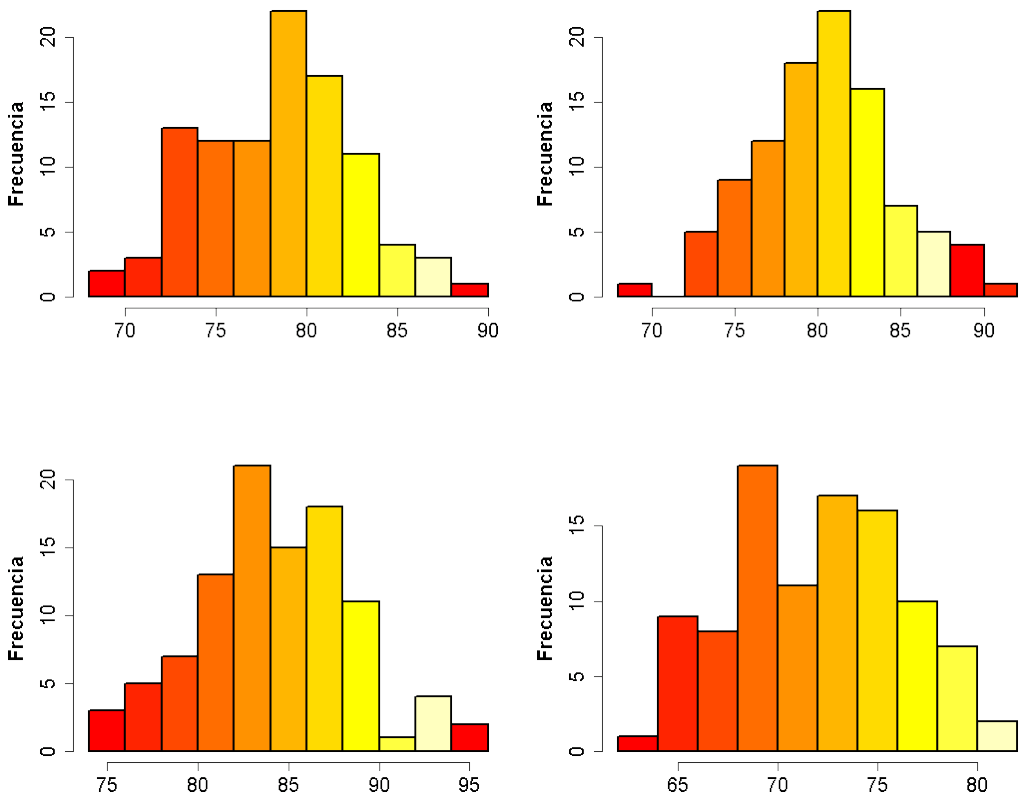

Figura 2: Histogramas paralelos del Ejemplo 11.1.1 del libro.

Dicho de otra manera, no hay ninguno que sea escandalosamente no normal. Y muchas veces nos tendremos que conformar con eso y con otra información adicional que nos permita suponer que las poblaciones son normales. Por ejemplo, ya hemos comentado que muchos fenómenos naturales se ajustan a la distribución normal y podemos usar la intuición de que el fenómeno que estudiamos es uno de ellos. En términos generales, los cuatro histogramas tienen forma de campana, con aproximadamente el mismo perfil.

- Las dos líneas con `par` que rodean a la línea con `tapply` sirven para fijar parámetros del gráfico. En concreto, la primera usa el parámetro, de aspecto algo cabalístico,

  ```
  mfrow=c(k%/%2+k%%2,2)
  ```

  para crear una *matriz de gráficos*. En el ejemplo concreto que estamos usando, los gráficos son cuatro, y podríamos colocarlos en una matriz 2×2 con `mfrow=c(2,2)`. Pero, puesto que queremos un código que funcione en el caso general, tendremos `k` niveles, y para colocarlos en una matriz de dos columnas tenemos que averiguar el número de filas. Eso es lo que hacen esas instrucciones misteriosas. Concretamente, `k%/%2` es el cociente de la división de k entre 2, mientras que `k%%2` es el resto. De esa forma, si tenemos, por ejemplo, $k = 9$ niveles, al ser $k = 2 \cdot 4 + 1$, sabremos que necesitamos $cociente + resto = 4 + 1 = 5$ filas para los nueve gráficos (con dos columnas, recuerda). El resto de los valores que aparecen en el primer `par` son simplemente "decorativos": tamaños de letra, tipografía, etiquetado, etc. El `mfrow=c(1,1)` que aparece en el segundo par sirve para devolver los gráficos de R a su estado anterior (una matriz 1×1 de gráficos).

- En la llamada a `tapply` la parte esencial es

  ```
  tapply(datos$Respuesta,datos$Tratamiento,hist,breaks=10,
  ```

  El resto es casi todo, de nuevo, "decorativo". Hemos pasado a `hist` la opción `breaks=10` para que todos los histogramas tuvieran el mismo número de columnas, lo que ayuda a comprobar si sus perfiles son similares.

Análisis de los residuos.

Como sabemos por nuestra experiencia en el modelo de regresión lineal el análisis de las condiciones de aplicabilidad del contraste Anova pasa por un análisis de los residuos. En esta incursión tan breve que estamos haciendo, nos vamos a limitar al análisis gráfico de los residuos, usando las herramientas que R nos proporciona. En particular, vamos a usar una nueva función, la función `aov` (de *analysis of variance*), que entre otras cosas nos va a permitir obtener ese análisis gráfico de los residuos. Para que R calcule los residuos y prepare esos gráficos, empezamos haciendo:

```
########################################
# Gráficos para el análisis mediante
# residuos de las hipótesis del modelo.
########################################

datos.aov = aov(datos$Respuesta ~ datos$Tratamiento, datos)
par(mfrow=c(2,2), cex.axis=1.5, lwd=2, font.lab=1.5)
plot(datos.aov)
par(mfrow=c(1,1))
```

El resultado se muestra en la Figura 3. Ya sabemos cuál es la misión de las dos llamadas a `par` que rodean a la función `plot`. Con independencia del número de niveles R siempre produce cuatro gráficos para el análisis de los residuos, así que hemos decidido colocarlos en una matriz de gráficos 2×2, usando `mfrow` dentro de `par`.

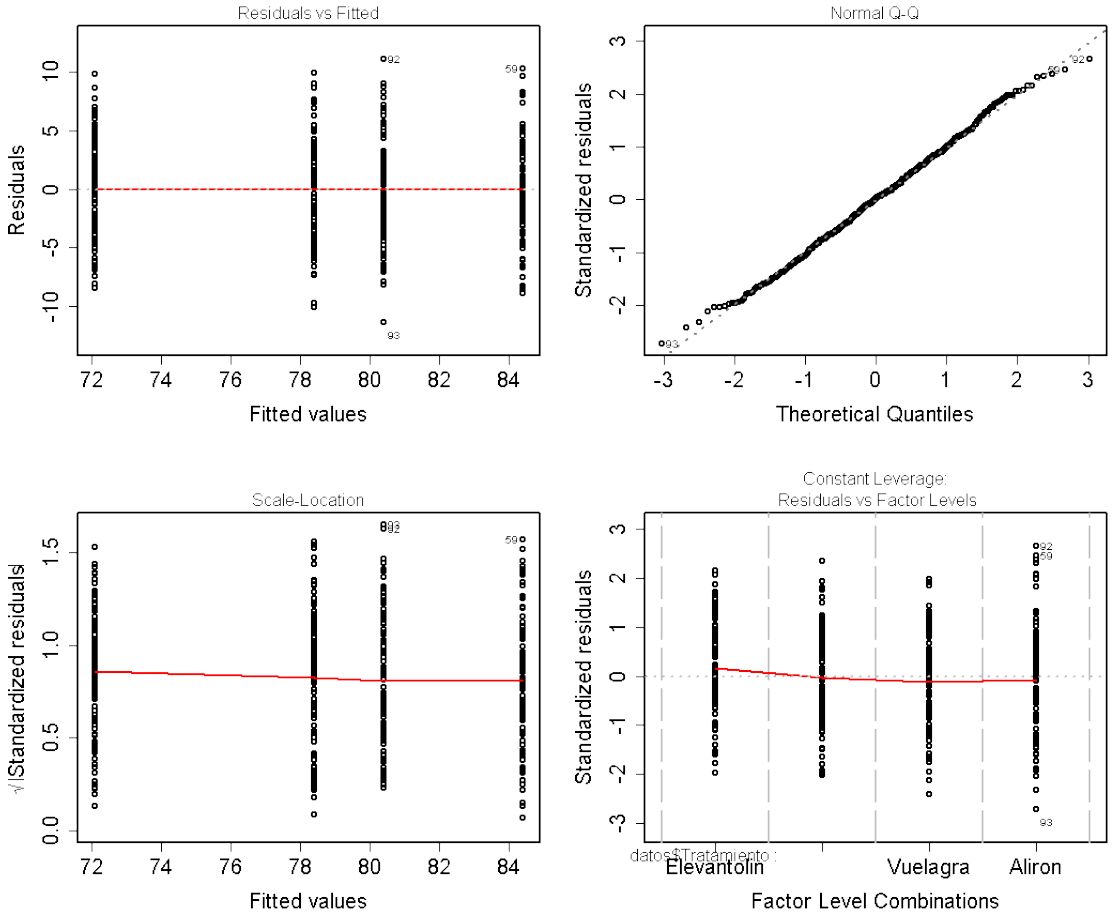

Figura 3: Análisis gráfico de los residuos para el Ejemplo 11.1.1 del libro.

De esos cuatro gráficos, vamos a centrar nuestra atención en los dos de la primera fila.

- El de la izquierda muestra los residuos frente a los valores que predice el modelo. Esos valores

predichos, en el Anova unifactorial, no son más que las medias de cada uno de los niveles; cuatro, para este ejemplo concreto. Por esa razón vemos cuatro grupos de puntos, con cada grupo situado, sobre el valor en el eje horizontal de cada una de las medias. Lo que debemos buscar, en este primer gráfico, para evaluar las condiciones de aplicabilidad de Anova, es si existe algún patrón apreciable en las distribuciones de los residuos, de forma que, por ejemplo, la dispersión de los residuos pudiera aumentar de izquierda a derecha (o viceversa). Es decir, buscamos la existencia de posibles patrones en forma de cuña, en el mismo sentido que vimos en la discusión en torno a la Figura 10.20 (pág. 390) del libro, para el caso del modelo de regresión lineal. Hay que tener en cuenta, por supuesto, que allí la variable explicativa era continua, mientras que aquí es discreta, y que eso se traduce en que los residuos se agrupan en las trazas verticales que hemos comentado, una por nivel.

- El segundo gráfico de la primera fila es un `qqplot` de los residuos, que se usa para obtener una nueva verificación de la hipótesis de normalidad de los residuos. En este caso, los residuos se ajustan muy satisfactoriamente a la recta, así que no parece que debamos preocuparnos por esa condición.

Naturalmente, hay mucho más que decir sobre este tema. Más allá de un análisis gráfico, existen contrastes formales que pueden utilizarse para verificar las condiciones de Anova. Y no hemos hecho, por ejemplo, una discusión sobre la posible existencia de valores atípicos o influyentes, similar a la que hicimos en el caso de la regresión lineal. En el Apéndice A del libro (pág. 567) se pueden encontrar referencias a tratamientos más avanzados de Anova.

Ejercicio 4. *Aplica el fichero* `Tut11-Anova-Avanzado.R` *para construir y analizar los modelos Anova que hemos construido en el Ejercicio 1, pág. 6. ¿Qué conclusiones obtienes sobre la validez de estos modelos?* □

3. Comparaciones dos a dos a posteriori

Una vez que sabemos que el contraste Anova ha sido significativo, y que, por tanto, las medias de los diferentes niveles no son todas iguales, la pregunta natural es: ¿qué parejas son significativamente diferentes?

Hay, de nuevo, mucho más que decir sobre este tema de lo que vamos a presentar aquí. Por ejemplo, una situación frecuente es aquella en la que existe un grupo (denominado grupo de control o, simplemente, control) y queremos comparar la media de los demás grupos con la de ese grupo de control. Este tema está íntimamente relacionado con el diseño del experimento subyacente a este modelo Anova, y para tratarlo con más detalle deberíamos entrar en esa discusión y en la noción de matriz de contraste, de la que hablaremos en la Sección 5. Aquí nos vamos a limitar a presentar las ideas más básicas sin entrar a fondo en la discusión del diseño del experimento. Como hemos dicho anteriormente, el Apéndice A del libro (pág. 567) contiene referencias a tratamientos completos del Diseño Experimental.

En R, la herramienta más sencilla para obtener ese tipo de comparaciones dos a dos es la función `pairwise.t.test`. Al usarla, debemos tener en cuenta que es necesario reajustar los p-valores, como hemos comentado en la Sección 11.6, para evitar la posibilidad de cometer un error de Tipo I (un falso positivo, rechazando la hipótesis nula, que nos llevaría a asegurar que hay diferencias entre los niveles del tratamiento), debido simplemente a la repetición de un número alto de contrastes. Hay muchos métodos para realizar ese ajuste, cada uno con sus ventajas e inconvenientes. Aquí vamos a adoptar un enfoque muy clásico, y vamos a utilizar el método de Bonferroni, que hemos descrito en la Sección 11.6.1 del libro (pág. 444) y que multiplica cada p-valor por el número de comparaciones que se han realizado. Para hacer esto en R, con los datos del Ejemplo 11.1.1 de los frailecillos, usamos el siguiente código:

```
########################################
# Comparaciones a posteriori, dos a dos
########################################
# En caso de un Anova significativo, podemos comparar los grupos dos a dos
# Los p-valores de las comparaciones se obtienen con este comando:

(ptt=pairwise.t.test(datos$Respuesta, datos$Tratamiento,
                     p.adj="bonferroni", pool.sd=FALSE))
```

```
##
##  Pairwise comparisons using t tests with non-pooled SD
##
## data:  datos$Respuesta and datos$Tratamiento
##
##              Aliron Elevantolin Plumiprofeno
## Elevantolin  0.005  -           -
## Plumiprofeno <2e-16 1e-09       -
## Vuelagra     <2e-16 <2e-16      <2e-16
##
## P value adjustment method: bonferroni
```

Fíjate que la respuesta contiene seis p-valores, porque al trabajar con cuatro niveles tenemos

$$\binom{4}{2} = 6$$

posibles parejas para comparar. Por ejemplo, el p-valor (ajustado mediante el método de Bonferroni) de la comparación entre *Plumiprofeno* y *Elevantolín* aparece como `1e-09`. Como decíamos, podemos conseguir una información más detallada así:

```
ptt$p.value
```

```
##                  Aliron Elevantolin Plumiprofeno
## Elevantolin  5.4566e-03          NA           NA
## Plumiprofeno 7.6636e-19  1.0408e-09           NA
## Vuelagra     2.5313e-20  1.4943e-30    1.148e-50
```

Y así vemos los p-valores con más precisión. Realmente, muchas veces no es necesario hacer esto, porque a menudo sólo nos interesa la magnitud del p-valor, y no necesitamos conocerlo con precisión. No obstante, esta vez lo hemos hecho así para que resulte más fácil ver la corrección de Bonferroni en acción. Hagamos una de esas seis comparaciones individuales por separado, usando el estadístico Υ que aparece en la Ecuación 11.19 del libro (pág. 444):

$$\Upsilon = \frac{\bar{X}_{\bullet i} - \bar{X}_{\bullet j}}{\sqrt{s^2_{\text{pond}} \cdot \left(\frac{1}{n_i} + \frac{1}{n_j}\right)}},$$

donde s^2_{pond} es la *cuasidesviación ponderada* (en inglés, *pooled sd*), que aparece en la Ecuación 11.20 del libro (pág. 444) y que se definide mediante

$$s^2_{\text{pond}} = \frac{SS_{\text{residual}}}{N - k}$$

Concretamente, vamos a usar esto para calcular el p-valor para el contraste de igualdad de medias entre *Plumiprofeno* y *Elevantolín*. Recuerda que el número de observaciones en cada nivel del factor está almacenado en el vector `replicas` de R. El cálculo es este:

```
(NumEstadistico = mediasPorNivel[2] - mediasPorNivel[3]  )
```

```
## Elevantolin
##     -4.0004
```

```
(pooledSd = sqrt(SSresidual/(N-k)))
```

```
## [1] 4.1997
```

```
(denomEstadistico = sqrt(pooledSd^2 * ((1 / replicas[2]) + (1 / replicas[3]))))
```

```
## Elevantolin
##     0.59393

(Estadistico = NumEstadistico / denomEstadistico)

## Elevantolin
##     -6.7355
```

No te preocupes por el nombre que aparece sobre el valor del estadístico, R lo arrastra porque `replicas` es una tabla con nombres. Ahora podemos usar la Ecuación 11.21 del libro (pág. 445) para calcular el p-valor con el ajuste de Bonferroni:

```
(NumComp = choose(k, 2))

## [1] 6

(pV = min(NumComp * 2 * (1 - pt(abs(Estadistico), df = N - k)), 1))

## [1] 3.4608e-10
```

Y comparamos con el resultado que obtuvimos en `pairwise.t.test` para esa pareja de niveles, para comprobar que coincide con nuestro cálculo "manual":

```
ptt$p.value[2, 2]

## [1] 1.0408e-09
```

El método de Bonferroni es, como hemos dicho, sólo una de las posibilidades. Su mayor limitación estriba en el hecho de que es un método de los denominados "conservadores". Es decir, que sacrifica excesivamente la potencia (en el sentido del Capítulo 7.3), con el fin de evitar los falsos positivos. Pero tiene la virtud, como hemos puesto de manifiesto, de ser conceptualmente muy sencillo y fácil de aplicar.

A la luz de estos resultados puede ser una buena idea regresar a la Figura 1 (pág. 7), en la que hemos mostrado los diagramas de caja biselados de este mismo ejemplo, y tratar de ver si los p-valores que hemos obtenido coinciden con las conclusiones que hubiéramos podido extraer de ese análisis gráfico.

Ejercicio 5. *Usa código como el que hemos visto en esta sección para comprobar los resultados que aparecen en los Ejemplos 11.6.4 (pág. 447) y 11.6.5 (pág. 448) del libro.* □

Representaciones gráficas para las comparaciones entre grupos.

Vamos a ver ahora el código necesario para obtener una representación gráfica como la de la Figura 11.4 del libro (pág. 450), que permite identificar fácilmente cuáles de las difernecias entre las medias de los niveles han resultado significativamente diferentes. Además, el código que incluimos permite obtener una representación alternativa de esa situación, como la de la Figura 4

No vamos a entrar en detalles sobre este código, porque eso nos ocuparía demasiado espacio. Remitimos al lector interesado a la documentación de la librería `multcomp`. El código aparece en el fichero `Tut11-Anova-Avanzado.R` del que ya hemos hablado, dentro de un bloque titulado *Comparaciones a posteriori, representaciones gráficas*. Para utilizar esa parte del fichero hay que hacer lo siguiente:

- Para empezar, por supuesto, fijar el directorio de trabajo y el nombre del fichero de datos.
- Asegurarnos de que hemos instalado las librerías `multcomp` y `multcompView` de R.
- Descomentar las líneas de código del bloque de comandos *Comparaciones a posteriori, representaciones gráficas*.

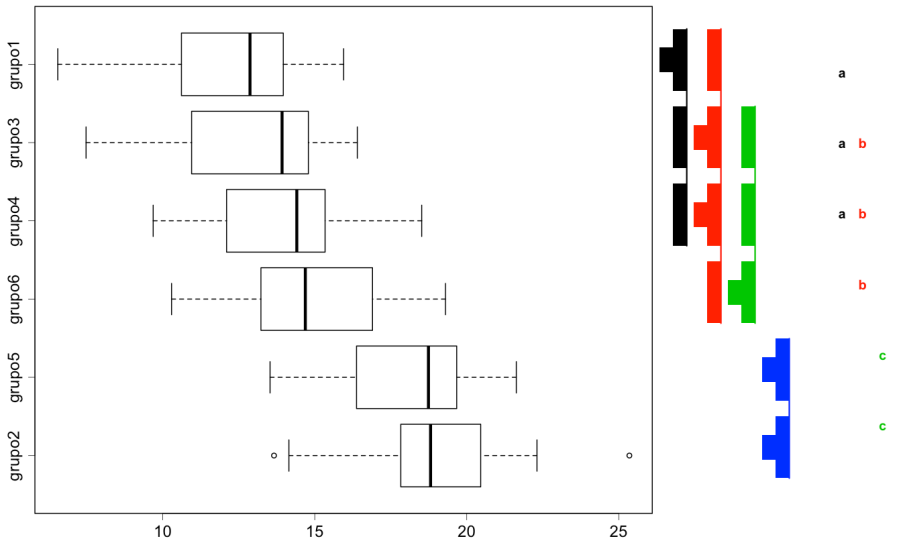

Figura 4: Otra representación de las comparaciones dos a dos para el Ejemplo 11.6.4 del libro.

Ejercicio 6. *Haz esto con los datos de los Ejemplos 11.6.4 (pág. 447) y 11.6.5 (pág. 448) del libro.* □

Comentarios sobre los denominados bargraphs

Los contrastes de diferencias de medias como el Anova se acompañan a menudo, en las publicaciones científicas, de diagramas de columnas con barras de error (*bargraphs with error bars*, en inglés), como el que aquí hemos incluido en la Figura 5. Estos diagramas se obtienen con mucha (demasiada) facilidad en las hojas de cálculo, lo cual sin duda explica parte de su popularidad. En el libro, concretamente en la página 311 hemos desaconsejado con insistencia el uso de este tipo de gráficos. Y ahora queremos volver a argumentar los problemas que merman la efectividad de estas gráficas para transmitir información veraz relativa a las diferencias entre las medias de los distintos tratamientos.

Las denominadas barras de error son esas figuras con forma de I que aparecen en la parte superior de cada columna. Uno de los problemas con este tipo de gráficas es que las barras de error ni siquiera representan intervalos de confianza para la media de cada uno de los grupos, por lo que no es correcto usarlas para concluir diferencias significativas entre las medias de esos grupos. Además, no tiene mucho sentido usar una columna para indicar el valor de una media. Como vimos en en el primer capítulo del curso, los diagramas de columnas son adecuados para representar *frecuencias*, y su uso aquí resulta confuso. A menudo el nivel base de las columnas no tiene ningún significado especial, y fijarlo arbitrariamente en 0 puede ser simplemente eso, una arbitrariedad, que además puede tener un impacto negativo en la capacidad del gráfico para transmitir la información relevante. Recomendamos, como alternativa, gráficos basados en intervalos de confianza, como el que aparece en la Figura 171, pág. 135 del libro *Introductory Statistics with R*, de Peter Dalgaard (ver la referencia en la Bibliografía del libro), y que son similares al que aquí incluimos en la Figura 6. En cualquier caso, El fichero `Tut11-Anova-Avanzado.R` contiene código para generar ambos tipos de gráficos.

4. ¿Y si mis datos no están en el formato correcto?

¡Ah, los placeres del *esquilado* de datos!

En inglés, se habla de data wrangling, donde el *wrangler* es esa imagen icónica del vaquero en el rodeo, volando sobre un caballo indómito... La verdad es que no soy capaz de identificar, en el trabajo que a menudo tenemos que hacer con los datos, nada de la épica a la que supuestamente alude esa terminología en inglés. El trabajo, mucho más a menudo, me recuerda a ese otro trabajo

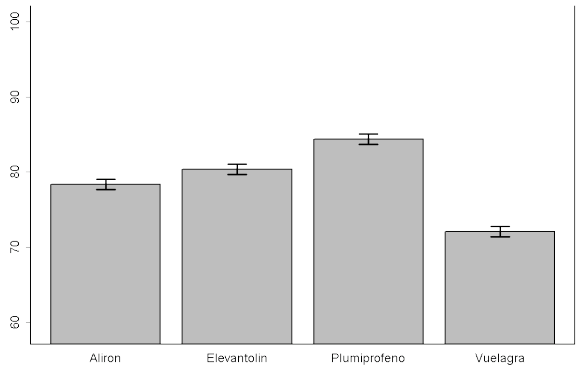

Figura 5: Diagrama de columnas con barras de error para el Ejemplo 11.1.1 del libro.
¡NO SE RECOMIENDA USAR ESTE TIPO DE GRÁFICOS!

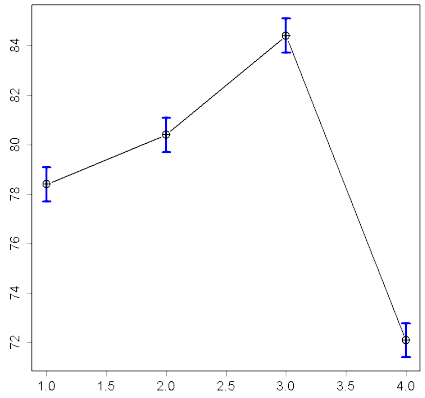

Figura 6: Un diagrama alternativo para el Ejemplo 11.1.1 del libro.

ganadero, desde luego menos épico, que consiste en agarrar a una oveja (en general, hostil a lo que pretendemos de ella) y bregar contra el animal y su pelaje hasta reducirlo a algo mucho más aseado y presentable. Por esa razón, modestamente, propongo el uso del verbo *esquilar* para el tipo de operaciones que nos vamos a ver llevados a acometer demasiado a menudo.

En el caso que nos ocupa, más temprano que tarde nos tropezaremos con una tabla como la "defectuosa" Tabla 11.1 (pág. 11.1) que hemos visto al comienzo del Capítulo 11 del libro. Se trata de "datos sucios", en el sentido que hemos discutido al comienzo de este tutorial, y nos toca a nosotros la tarea de adecentarlos. No podemos ofrecerle al lector una solución general al problema, porque, por su propia esencia, los datos sucios existen en una pléyade de variantes, y no se pliegan a un tratamiento uniforme. Pero sí podemos mostrarle una estrategia que funciona, al menos en el caso de tablas como la que nos ocupa. Y, desde luego, es un hecho constatado que se aprende con la experiencia y que, a medida que el lector se vaya encontrando en el futuro con otras dificultades, los métodos, técnicas y trucos del oficio de esquilador de datos le servirán para fabricar una solución a medida del problema.

Sin más preámbulos, vamos a presentar el fichero

Tut11-Anova-EsquilandoDatos.R

Las instrucciones y comentarios de ese fichero de código deberían servir de explicación suficiente del funcionamiento del fichero. Su propósito es leer un fichero de texto plano con datos (típicamente un fichero `csv`), en el que se ha guardado una tabla como la Tabla 11.1 que hemos mencionado, y producir como resultado un nuevo fichero csv con la versión "limpia" de esos datos, lista, por ejemplo, para usarla con los ficheros que hemos visto en las dos primeras secciones de este tutorial.

Un ejemplo típico de tabla de entrada ("sucia") puede ser este:

Lago1	Lago2	Lago3	Lago4
7.1	7.2	5.6	7.2
8.5	6.5	7.1	6.6
6.2	5.9	6.3	6.3
7.3	7.8	6.7	7.4
7.9		6.5	

que están disponibles en el fichero

Tut11-DatosSucios.csv

El aspecto más complicado de ese fichero es que es necesario contemplar la posibilidad de que, como en esta tabla, haya un número distinto de réplicas para cada uno de los niveles del tratamiento. Por eso la función `read.table` incluye un argumento `na.strings=c("NA",,)` (del inglés, *Not Available*), cuyo propósito es detectar las posibles formas de codificar un dato "ausente" en la tabla. La gestión adecuada de esos datos ausentes es, como hemos comentado en alguna otra ocasión, una de las dificultades más comunes del análisis de datos.

5. Introducción a los contrastes con R.

Opcional: esta sección puede omitirse en una primera lectura.

En la Sección 1.3 (pág. 8) hemos usado la función `lm` para obtener de manera rápida la tabla Anova del ejemplo de los frailecillos. Recordemos que esa tabla era:

```
##
## Call:
## lm(formula = Respuesta ~ Tratamiento, data = datos)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11.37  -2.81   0.11   2.77  11.11
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                78.399      0.420  186.68  < 2e-16 ***
## TratamientoElevantolin      2.000      0.594    3.37  0.00083 ***
## TratamientoPlumiprofeno     6.001      0.594   10.10  < 2e-16 ***
## TratamientoVuelagra        -6.299      0.594  -10.61  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.2 on 396 degrees of freedom
## Multiple R-squared:  0.531,Adjusted R-squared:  0.527
## F-statistic:  149 on 3 and 396 DF,  p-value: <2e-16
```

En aquella sección vimos que el primer valor que aparece en la fila `Intercept` es la media $\bar{X}_{\cdot 1} = 78.40$ del primer nivel (por orden alfabético), el *Aliron*. Los valores iniciales de las siguientes tres filas son *las diferencias* entre este valor y las medias de los restantes tres niveles:

$$\begin{cases} \bar{X}_{\cdot 2} - \bar{X}_{\cdot 1} = 80.40 - 78.40 = 2.00 \\ \bar{X}_{\cdot 3} - \bar{X}_{\cdot 1} = 84.40 - 78.40 = 6.00 \\ \bar{X}_{\cdot 4} - \bar{X}_{\cdot 1} = 72.10 - 78.40 = -6.30 \end{cases}$$

También dijimos entonces que íbamos a tratar de entender por qué nos pueden interesar estos valores, y qué significan las restantes columnas de la tabla en la que aparecen. Además, en esta sección vamos a aprender a controlar el proceso, para que los cálculos de R se ajusten a nuestros deseos en una situación experimental concreta.

A la vista de la tabla está claro que:

- Los coeficientes de la primera columna están relacionados con los coeficientes $b_0, b_1, \ldots, b_4$ del modelo lineal Anova que aparecen en el Ejemplo 11.4.3 del libro (pág. 435). Por ejemplo,

$$b_2 - b_1 = \left(\bar{X}_{\cdot 2} - b_0\right) - \left(\bar{X}_{\cdot 1} - b_0\right) = \bar{X}_{\cdot 2} - \bar{X}_{\cdot 1}$$

- La suma de los coeficientes b_1, b_2, b_3, b_4 es igual a 0, por su propia definición (y si en el el Ejemplo 11.4.3 la suma *parece* no ser cero es por los efectos del redondeo). En términos prácticos, eso significa que uno de esos coeficientes *sobra, está de más*. No aporta ninguna información que no podamos extraer de los restantes. En la Sección 5 del Tutorial veremos que la forma en la que R ha expresado el modelo tiene, entre otras, la virtud de que *no hay coeficientes de sobra*. Por eso la tabla `Coefficients` en la salida de `summary(datos.lm)` tiene sólo cuatro filas, mientras que $b_0, \ldots, b_4$ son cinco coeficientes.

La idea clave para controlar la información que aparece en la tabla Anova es la noción de contraste que se discute en la Sección 11.6.2 del libro (pág. 453). Vamos a aprender cómo usa R esos contratses y, en particular, cómo definir nuestros propios contrastes. Pero para llegar hasta ahí el primer paso es entender en detalle la noción de variables *dummy* que hemos visto en la página 431 del libro.

Variables dummy en R, construcción a mano.

Al leer la Sección 11.4 del libro (pág. 430) es posible que te hayas preguntado cómo construir en R esas variables índice (o variables *dummy*) que hemos utilizado para expresar Anova como un modelo lineal. Lo primero que queremos apresurarnos a decir es que en realidad no es necesario hacer esto: las variables de tipo `factor` de R sirven, entre otras cosas, para que no tengamos que ocuparnos de crear estas variables índice manualmente. Podemos relajarnos y dejar que R se encargue de ese trabajo por nosotros.

Pero, como suele suceder, mientras uno está aprendiendo a hacer algo por primera vez, esa construcción manual puede ayudar a entender lo que estamos haciendo. Es una situación parecida a la construcción manual de la tabla Anova que hemos hecho antes. Por supuesto, los profesionales usan siempre `lm` para obtener esa tabla. Pero es bueno, pedagógicamente, construirla a mano las primeras veces. Una de las razones por las que nos gustan las herramientas como R es que, a diferencia de soluciones tipo *caja negra*, permiten hacer ese tipo de construcciones manuales con tanto detalle como se desee. Además, como hemos dicho, entender las variables dummy nos va a facilitar mucho la tarea de entender como se gestionan los contrastes en R.

Manos a la obra entonces. Vamos a construir las variables T_1, T_2, T_3, T_4 del Ejemplo 11.4.1 del libro (pág. 432). La idea es muy sencilla: para definir, por ejemplo, una variable `T1` que valga 1 en los frailecillos tratado scon el primer tratamiento (*Alirón*) basta con usar este código:

```
T1  = as.integer(datos$Tratamiento == "Aliron")
```

Hemos usado la función `as.integer` para convertor los valores booleanos `TRUE/FALSE` en 0s y 1s. Ya sabes que R los gestiona así directamente, pero esta vez queríamos hacer explícita la codificación numérica. Un inconveniente de este código es que depende del nombre concreto de ese nivel. Podemos hacerlo mejor definiendo así las variables índice:

```
### Variables indice

T1  = as.integer(datos$Tratamiento == levels(datos$Tratamiento)[1])
T2  = as.integer(datos$Tratamiento == levels(datos$Tratamiento)[2])
T3  = as.integer(datos$Tratamiento == levels(datos$Tratamiento)[3])
T4  = as.integer(datos$Tratamiento == levels(datos$Tratamiento)[4])
```

Por lo demás, si fuéramos a escribir un código general para este tipo de operaciones deberíamos además escribirlo de forma que el código se adaptara de forma automática al número de niveles del factor tratamiento. Pero como sólo estamos haciendo un experimento nos vamos a conformar con el anterior método para definir las T_i.

Podemos añadir estas variables al `data.frame` con los datos usando `cbind`, como si fuera una matriz:

```
datos = cbind(datos, T1, T2, T3, T4)
```

Y si observamos las primeras 10 filas de los datos veremos que las variables índice que hemos construido hacen exactamente lo que esperábamos de ellas:

```
head(datos, 10)
```

```
##     Tratamiento Respuesta T1 T2 T3 T4
## 1        Aliron     76.65  1  0  0  0
## 2   Elevantolin     88.66  0  1  0  0
## 3        Aliron     79.36  1  0  0  0
## 4      Vuelagra     76.74  0  0  0  1
## 5        Aliron     71.83  1  0  0  0
## 6      Vuelagra     74.72  0  0  0  1
## 7  Plumiprofeno     87.14  0  0  1  0
## 8        Aliron     73.24  1  0  0  0
## 9   Elevantolin     78.12  0  1  0  0
## 10 Plumiprofeno     82.34  0  0  1  0
```

Es importante entender que estas variables índice contienen exactamente la misma información que la variable `Tratamiento`, pero codificada de otra manera. Antes, para construir el modelo lineal del Anova hemos usado la función `lm` con esta fórmula:

```
datos.lm = lm(Respuesta ~ Tratamiento, data = datos)
```

Pero, puesto que las variables índice contienen la misma información que `Tratamiento`, podemos usarlas como variables predictoras para construir el modelo. No podemos entrar en todos los detalles aquí, ni mucho menos, porque eso supondría adentrarnos en el terreno de los modelos con más de una variable predictora. Pero podemos decir que cuando se usan estos modelos en R, la forma más sencilla de indicar que hay varias variables en el modelo es separándolas con el símbolo `+`, como hemos hecho en este nuevo intento de construir el modelo:

```
(datos.lm2 = lm(Respuesta ~ T1 + T2 + T3 + T4, data = datos))
```

```
##
## Call:
## lm(formula = Respuesta ~ T1 + T2 + T3 + T4, data = datos)
##
## Coefficients:
## (Intercept)           T1           T2           T3           T4
##        72.1          6.3          8.3         12.3           NA
```

De nuevo una pregunta parecida a la de antes: ¿qué son esos cinco coeficientes? La situación se ve agravada además por el hecho de que uno de ellos es `NA`. El problema se debe, como hemos señalado, al hecho de que estamos tratando de usar cuatro variables explicativas *que no son independientes*; sobra una de ellas. Si sabes algo de Álgebra Lineal, la situación es similar a tratar de resolver un sistema de ecuaciones lineales en el que la matriz de coeficientes tiene determinante cero: lo normal es que tengamos problemas para hacerlo. R, en este caso, al constatar que sobraban variables ha tomado como término independiente del modelo la media del cuarto nivel del tratamiento, $\bar{X}_{\cdot 4} \approx 72.1$ y ha trabajado con una ecuación del modelo de la forma:

$$X = c_0 + c_1 \cdot T^{(1)} + c_2 \cdot T^{(2)} + c_3 \cdot T^{(3)}$$

en la que además,

$$\begin{cases} c_0 = \bar{X}_{\cdot 4} \approx 72.1 \\ c_1 = \bar{X}_{\cdot 1} - \bar{X}_{\cdot 4} \approx 6.299 \\ c_2 = \bar{X}_{\cdot 2} - \bar{X}_{\cdot 4} \approx 8.3 \\ c_3 = \bar{X}_{\cdot 3} - \bar{X}_{\cdot 4} \approx 12.3 \end{cases}$$

En este caso hemos dejado que R, al comprobar que sobran variables explicativas, eligiera cuál de esas variables se eliminaba. Podemos controlar un poco más el proceso, eligiendo nosotros mismos una de las variables índice para descartarla y construir el modelo a partir de las restantes. Por ejemplo, si decidimos descartar `T1`, entonces hacemos:

```
(datos.lm3 = lm(Respuesta ~ T2 + T3 + T4, data = datos))

##
## Call:
## lm(formula = Respuesta ~ T2 + T3 + T4, data = datos)
##
## Coefficients:
## (Intercept)           T2           T3           T4
##        78.4          2.0          6.0         -6.3
```

Si ahora comparas estos resultados con los que obtuvimos al usar inicialmente `lm` así:

```
datos.lm = lm(Respuesta ~ Tratamiento, data = datos)
```

y que, recordémoslo, eran:

```
datos.lm$coefficients

##             (Intercept)  TratamientoElevantolin TratamientoPlumiprofeno
##                 78.3993                  2.0004                  6.0008
##     TratamientoVuelagra
##                 -6.2994
```

parece que hemos empezado a entender la respuesta inicial de R. Decimos que hemos empezado a entenderla porque el contexto natural de esta discusión es el lenguaje de contrastes.

5.1. Contrastes en R.

En la Sección 11.6.2 del libro (pág. 453) hemos visto que la técnica de los *contrastes* puede usarse como forma de adaptar el método Anova a algunas peculiaridades de nuestro diseño experimental. Por ejemplo en presencia de un grupo (nivel) especial de tratamiento que queremos comparar con los demás. Los *contrastes* son también una respuesta al problema de la dependencia de las variables índice (*dummy*).

Con R es muy fácil usar este método. Empecemos recordando que en la práctica un contraste se define mediante una combinación lineal de las medias de los distintos niveles del factor

$$a_1 \cdot \mu_1 + a_1 \cdot \mu_1 + \cdots + a_k \cdot \mu_k$$

con la condición de que la suma de los coeficientes a_i es igual a 0:

$$a_1 + a_2 + \cdots + a_k = 0.$$

Para usarlos en un contraste tipo Anova debemos seleccionar una colección de $k-1$ contrastes independientes, identificados mediante una matriz. Cuando usamos la función `lm` para hacer un contraste Anova, R selecciona por defecto una cierta *matriz de contrastes*. Para ver cuál es esa matriz vamos a empezar usando este comando:

```
contrasts(datos$Tratamiento)

##              Elevantolin Plumiprofeno Vuelagra
## Aliron                 0            0        0
## Elevantolin            1            0        0
## Plumiprofeno           0            1        0
## Vuelagra               0            0        1
```

Esta no es todavía la matriz de contrastes, pero está estrechamente relacionada con ella. ¿Qué representa esta matriz, cómo se interpretan esos números? Para entenderlo, vamos a recordar que, como hemos visto en la Sección 11.6.2, hay dos formas de caracterizar un conjunto de contrastes. Una es mediante la *matriz del contraste* M que interviene en:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_k \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,k} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,k} \\ & & \ddots & \\ a_{k,1} & a_{k,2} & \cdots & a_{k,k} \end{pmatrix} \cdot \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_k \end{pmatrix}$$

La otra es mediante su inversa M^{-1}, que es la más cómoda cuando se trata de construir las variables índice asociadas a ese conjunto de contrastes. La matriz que se obtiene con `contrasts` está relacionada con esa inversa M^{-1} y eso hace que, al principio, sea un poco más difícil reconocer el conjunto de contrastes que está usando R. Para obtener la matriz de contrastes, como hemos argumentado en el libro (ver pág. 460), tenemos que añadir a la izquierda una columna de unos y calcular la inversa de la matriz resultante. En R la inversa de una matriz se calcula con `solve` (ver el Tutorial03):

```
(Minversa = cbind(rep(1, k),contrasts(datos$Tratamiento)))

##                Elevantolin Plumiprofeno Vuelagra
## Aliron       1           0            0        0
## Elevantolin  1           1            0        0
## Plumiprofeno 1           0            1        0
## Vuelagra     1           0            0        1

(M = solve(Minversa))

##              Aliron Elevantolin Plumiprofeno Vuelagra
##                   1           0            0        0
## Elevantolin      -1           1            0        0
## Plumiprofeno     -1           0            1        0
## Vuelagra         -1           0            0        1
```

Esta última matriz M es la que en el libro hemos llamado la matriz del contraste. Y, ahora sí, nos permite ver claramente el conjunto de contrastes queR ha decidido usar:

$$\begin{cases} \alpha_1 = \mu_1, \\ \alpha_2 = \mu_2 - \mu_1, \\ \alpha_3 = \mu_3 - \mu_1, \\ \alpha_4 = \mu_4 - \mu_1. \end{cases} \Rightarrow M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

¿Por qué este conjunto? Porque la opción por defecto consiste en suponer que queremos comparar un *nivel especial* frente a todos los demás niveles. Por ejemplo puede ser un placebo que se compara con varios tratamientos, o un tratamiento ya establecido frente a varias alternativas nuevas o viceversa, etc. Y a falta de más información, R usa como *nivel especial* el primer nivel (que, en este caso, es *Alirón* porque los niveles se han ordenado alfabéticamente).

Para completar la información, R va a usar 3 variables índice, a las que vamos a llamar $\tilde{T}^{(1)}$, $\tilde{T}^{(2)}$ y $\tilde{T}^{(3)}$ (R las nombra como algunos de los tratamientos por razones que después quedarán claras), y que se definen mediante una tabla, cuyas columnas son las tres últimas columnas de M^{-1}:

	$\tilde{T}^{(1)}$	$\tilde{T}^{(3)}$	$\tilde{T}^{(4)}$
Alirón:	0	0	0
Elevantolín:	1	0	0
Plumiprofeno:	0	1	0
Vuelagra:	0	0	1

Como hemos dicho, la tabla que define estas variables índice es precisamente la matriz que se obtiene usando `contrasts(datos$Tratamiento)`.

Interpretación de los resultados de `lm`.

Ahora que ya sabemos cuál es el conjunto de contrastes que R ha decidido usar estamos listos para entender con más detalle lo que se obtiene al usar la función `lm`, un trabajo que habíamos dejado pendiente desde el principio de la Sección 1.3 (pág. 8).

```
summary(lm(datos$Respuesta ~ datos$Tratamiento))
```

```
##
## Call:
## lm(formula = datos$Respuesta ~ datos$Tratamiento)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11.37  -2.81   0.11   2.77  11.11
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      78.399      0.420  186.68  < 2e-16 ***
## datos$TratamientoElevantolin      2.000      0.594    3.37  0.00083 ***
## datos$TratamientoPlumiprofeno     6.001      0.594   10.10  < 2e-16 ***
## datos$TratamientoVuelagra        -6.299      0.594  -10.61  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.2 on 396 degrees of freedom
## Multiple R-squared:  0.531,Adjusted R-squared:  0.527
## F-statistic:  149 on 3 and 396 DF,  p-value: <2e-16
```

Concretamente, empecemos por la columna `Estimate` de la tabla `Coefficients`. Recuerda que el modelo Anova que R está usando (el que elige por defecto) es:

$$X = \alpha_0 + c_1 \cdot T^{(1)} + c_2 \cdot T^{(2)} + c_3 \cdot T^{(3)} + \varepsilon$$

donde los α_i son los contrastes que hemos visto:

$$\alpha_1 = \mu_1, \quad \alpha_2 = \mu_2 - \mu_1, \quad \alpha_3 = \mu_3 - \mu_1, \quad \alpha_4 = \mu_4 - \mu_1$$

Los valores `Estimate` son las estimaciones que R ha obtenido de los valores de los α. Y debería ser evidente que, por ejemplo, la estimación de $\alpha_3 = \mu_3 - \mu_1$ sólo puede ser $\bar{X}_{\cdot 3} - \bar{X}_{\cdot 1}$. Comprobémoslo para todos los α_i (el caso de α_1 es especial, por eso lo hemos separado de los demás):

```
alphas = summary(datos.lm)$coefficients[, 1]

alphas[1]

## (Intercept)
##      78.399

mediasPorNivel[1]

## Aliron
## 78.399

alphas[2:4]

##  TratamientoElevantolin TratamientoPlumiprofeno     TratamientoVuelagra
##                  2.0004                  6.0008                  -6.2994

mediasPorNivel[2:4] - mediasPorNivel[1]

##  Elevantolin Plumiprofeno     Vuelagra
##       2.0004       6.0008      -6.2994
```

Como esperábamos, las estimaciones coinciden con las correspondientes diferencias de medias por niveles. Ahora podemos entender el resto de las columnas de esa tabla `Coefficients`. Lo que queremos saber, cuando utilizamos un cierto conjunto de contrastes, es cuáles de ellos son significativos. Por ejemplo, al usar el contraste $\alpha_2 = \mu_2 - \mu_1$, si ese contraste resulta significativo podremos concluir que hay una diferencia significativa entre las medias de los niveles 1 y 2 del tratamiento. Es decir, que la hipótesis nula que estamos contrastando con el contraste α_2 es:

$$H_0 = \{\alpha_2 = \mu_2 - \mu_1 = 0\}$$

El estadístico que necesitamos para este contraste (contraste de hipótesis sobre el contraste α_2, la terminología es así de confusa...) aparece en la Ecuación 11.28 del libro (pág. 459).

$$\Xi = \frac{\left(\sum_{j=1}^{k} a_j \cdot \bar{X}_{\cdot j}\right) - \left(\sum_{j=1}^{k} a_j \cdot \mu_j\right)}{\sqrt{s^2_{\text{pond}} \cdot \sum_{j=1}^{k} \frac{a_j^2}{n_j}}}$$

donde s^2_{pond} es la *cuasidesviación ponderada* (en inglés, *pooled sd*), que ya hemos calculado antes para usarla en el ajuste de Bonferroni y que hemos guardado en la variable `poolesSd` de R. Vamos a calcular entonces el estadístico Ξ para el contraste $\alpha_2 = \mu_2 - \mu_1$. Para hacerlo necesitamos además los valores n_k, el número de observaciones en cada nivel del factor, que hemos almacenado en el vector `replicas` (ver el bloque de *Construcción manual de la tabla Anova* en el fichero `Tut11-Anova-Basico.R`).

```
(numerEstad = t(mediasPorNivel) %*% M[2, ])

##         [,1]
## [1,] 2.0004

(denomEstad = pooledSd * sqrt(sum(M[2, ]^2 / replicas)))

## [1] 0.59393

(Estadistico = numerEstad / denomEstad)

##         [,1]
## [1,] 3.3681
```

Para entender esto necesitamos un poco de álgebra lineal. Hemos usado el producto matricial `%*%` del vector fila `t(mediasPorNivel)` por la segunda fila de la matriz de contrastes, para calcular el numerador del estadístico. Si sabes algo de álgebra, también puedes verlo como el producto escalar de los vectores $(a_1, \ldots, a_j)$ y $(\bar{X}_{\cdot 1}, \ldots, \bar{X}_{\cdot j})$

Este valor del estadístico aparece en la segunda fila de la columna titulada `t value` de la tabla `Coefficients`. Es la segunda fila porque estamos trabajando con el contraste α_2, claro. Y ahora podemos usar la distribución t de Student adecuada, con $N-k$ grados de libertad, para calcular el p-valor del contraste de hipótesis sobre el contraste α_2. Tengamos en cuenta que la hipótesis nula que estamos contrastando es:

$$H_0 = \{\alpha_2 = 0\}$$

así que se trata de un contraste bilateral:

```
(pValor = 2 * (1 - pt(abs(Estadistico), df=N-k)))

##             [,1]
## [1,] 0.00083106
```

y puedes comprobar que el resultado coincide con lo que aparece en la segunda fila de la última columna de la tabla `Coefficients`, titulada `Pr(>|t|)`. Por cierto, la segunda columna de esa tabla contiene, en sus tres últimas filas, el valor que nosotros hemos llamado `denomEstad`.

Ejercicio 7. *Comprueba, igual que hemos hecho aquí, el resto de valores de la tabla* `Coefficients`. □

Contrastes definidos por el usuario.

Acabamos de ver que cuando usamos la función `lm` para hacer un contraste Anova, R utiliza por defecto un cierto conjunto de contrastes. Concretamente, si tenemos k niveles del tratamiento, sean $T_1, T_2, \ldots, T_k$, con medias respectivas $\mu_1, \mu_2, \ldots, \mu_k$, entonces R usa:

$$\begin{cases} \alpha_2 = \mu_2 - \mu_1, \\ \alpha_3 = \mu_3 - \mu_1, \\ \quad \vdots \\ \alpha_k = \mu_k - \mu_1. \end{cases}$$

donde, como ya sabemos, α_1 no es un contraste. Este conjunto de contrastes es adecuado si consideramos que el nivel T_1 es especial en algún sentido y queremos compararlo con todos los restantes grupos. Pero hay muchas otras situaciones imaginables, que aparecen con frecuencia en el trabajo de los investigadores. Para dar cabida a esos objetivos experimentales, R pone a nuestra disposición la posibilidad de crear nuestra propia matriz de contrastes. Vamos a ver un ejemplo de cómo se hace esto.

Para empezar, vamos a fijar el objetivo que nos hemos planteado. En muchas ocasiones, los investigadores tienen una idea previa de cómo se ordenan las medias de los niveles del tratamiento. Imagínate que en nuestro ejemplo de los frailecillos, un investigador sospecha que la ordenación de las medias es esta:

$$\mu_4 < \mu_1 < \mu_3 < \mu_2$$

Otra forma de expresar esto es

$$0 < \mu_1 - \mu_4, \quad 0 < \mu_3 - \mu_1, \quad 0 < \mu_2 - \mu_3.$$

Para comprobar estas ideas, podemos diseñar esta colección de contrastes:

$$\begin{cases} \alpha_2 = \mu_1 - \mu_4 \\ \alpha_3 = \mu_3 - \mu_1 \\ \alpha_4 = \mu_2 - \mu_3 \end{cases}$$

Añadimos este término independiente:

$$\alpha_1 = \mu_4$$

¿Por qué elegimos este término independiente? Volveremos sobre esto enseguida. La matriz de este conjunto de contrastes es:

$$M = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

En R esto es:

```
(M = matrix(c(0, 0, 0, 1, 1, 0, 0, -1,-1, 0, 1, 0, 0, 1, -1, 0),
            nrow = 4, byrow = TRUE))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    1
## [2,]    1    0    0   -1
## [3,]   -1    0    1    0
## [4,]    0    1   -1    0
```

Y la inversa, que nos dará la tabla de valores índice, es:

```
(Minv = solve(M))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    0    0
## [2,]    1    1    1    1
## [3,]    1    1    1    0
## [4,]    1    0    0    0
```

Al eliminar la primera columna de unos, esto nos conduce a esta tabla de valores para las variables índice:

	$\tilde{T}^{(1)}$	$\tilde{T}^{(3)}$	$\tilde{T}^{(4)}$
Alirón:	1	0	0
Elevantolín:	1	1	1
Plumiprofeno:	1	1	0
Vuelagra:	0	0	0

Dejamos al lector la tarea de comprobar que, en efecto, con estas ecuaciones el modelo

$$X = \alpha_0 + c_1 \cdot T^{(1)} + c_2 \cdot T^{(2)} + c_3 \cdot T^{(3)} + \varepsilon$$

hace las predicciones correctas para cada uno de los cuatro grupos.

¿Cómo le pedimos entonces a R que use este conjunto de contrastes al aplicar la función `lm`? Primero tomamos la tabla de variables índice, eliminando la columna de unos inicial. Llamemos `indices` a esa tabla:

```
(indices = Minv[, 2:k])

##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    1    1    1
## [3,]    1    1    0
## [4,]    0    0    0
```

Y ahora usamos la misma función `contrasts` que vimos antes, pero en esta ocasión para asignarle como valor esta tabla de `indices`:

```
contrasts(datos$Tratamiento) = indices
```

Con eso estamos listos para invocar la función `lm`. Se obtiene:

```
summary(lm(datos$Respuesta ~ datos$Tratamiento, ))

##
## Call:
## lm(formula = datos$Respuesta ~ datos$Tratamiento)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -11.37  -2.81   0.11   2.77  11.11
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           72.100      0.420  171.68  < 2e-16 ***
## datos$Tratamiento1     6.299      0.594   10.61  < 2e-16 ***
## datos$Tratamiento2     6.001      0.594   10.10  < 2e-16 ***
## datos$Tratamiento3    -4.000      0.594   -6.74  5.8e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.2 on 396 degrees of freedom
## Multiple R-squared:  0.531,Adjusted R-squared:  0.527
## F-statistic:  149 on 3 and 396 DF,  p-value: <2e-16
```

Fíjate en que todos los contrastes han resultado significativos (con p-valores muy pequeños en la columna `Pr(>|t|)` de la tabla `Coefficients`). Pero fíjate además que la estimación que hemos obtenido de $\alpha_4 = \mu_2 - \mu_3$ es negativa. Es decir, que tenemos razones para pensar que, en realidad, lo

que se cumple es $\mu_2 - \mu_3 < 0$ o, lo que es lo mismo, $\mu_2 < \mu_3$. Cuando los contrastes son significativos, los signos de los coeficientes α pueden, por tanto, servir al investigador para comprobar si las medias de los niveles siguen el orden que habíamos conjeturado o, como sucede en este caso, alguna de las desigualdades es al revés de lo esperado.

6. Ejercicios adicionales y soluciones.

Ejercicios adicionales.

Ejercicio 8. *Un investigador está estudiando un nuevo elemento químico ultrapesado, del que sospecha que tiene cinco isótopos. Para comprobarlo ha aislado cinco muestras independientes, cada una con 50 mediciones de la masa atómica de uno esos posibles isótopos, a los que ha llamado $B1$, $B2$, $B3$, $B4$ y $B5$. El fichero adjunto*

tut11-isotopos.csv

contiene los datos que se han medido, en el formato adecuado. Haz un contraste Anova para confirmar que hay diferencias significativas entre las masas atómicas medias de esas cinco muestras. Es necesario que calcules el p-valor de ese contraste. Después, trata de comparar dos a dos las masas atómicas medias de las cinco muestras, explicando en detalle cómo lo haces. ¿Cuántos isótopos distintos crees que hay, en realidad? □

Fin del Tutorial11. ¡Gracias por la atención!