

# Diseños confundidos

## Ejemplo resuelto.

Una investigación busca mejorar la pureza de un producto químico, para ello se involucra la influencia de tres factores: Tasa de agitación ( $A$ ), concentración del compuesto base ( $B$ ) y concentración del reactivo ( $C$ ). El químico estableció un experimento con un diseño factorial con factores a dos niveles para un arreglo factorial  $2^3$ .

Se realizaron tres réplicas del experimento, pero solo podría realizar cuatro corridas del proceso químico en un día, esto llevó a que cada réplica debía correrse en dos días diferentes (bloques).

En la conducción del experimento, se construyó un diseño de bloques incompletos confundiendo la interacción de orden tres  $ABC$  con los bloques. En este caso el contraste de definición es  $L = x_1 + x_2 + x_3$  y el diseño se ilustra en la figura.

En la figura  $\circ$  representa las corridas con  $(ABC)_0$  y  $\bullet$  las corridas con  $(ABC)_1$

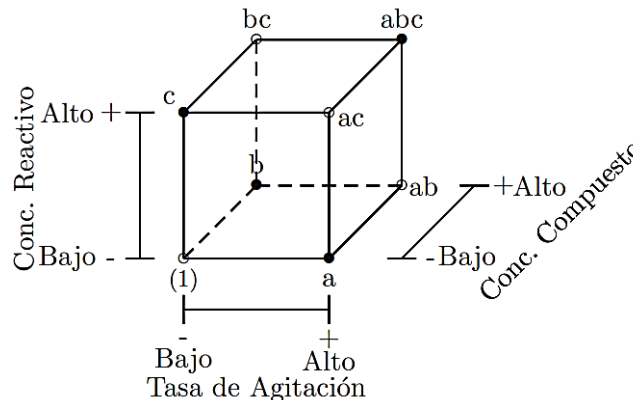


Figura 10.1. Vista geométrica del diseño factorial  $2^k$  en dos bloques.

Las combinaciones de tratamientos dentro de cada bloque se muestran en la tabla.

Réplica I		Réplica II		Réplica III	
Bloque 1 ( $ABC$ ) <sub>1</sub>	Bloque 2 ( $ABC$ ) <sub>0</sub>	Bloque 3 ( $ABC$ ) <sub>0</sub>	Bloque 4 ( $ABC$ ) <sub>1</sub>	Bloque 5 ( $ABC$ ) <sub>0</sub>	Bloque 6 ( $ABC$ ) <sub>1</sub>
001 : 44,5	000 : 46,8	101 : 49,8	001 : 55,5	011 : 53,2	100 : 69,5
010 : 44,2	011 : 44,5	110 : 52,0	100 : 59,8	101 : 57,2	010 : 62,8
100 : 60,1	101 : 57,0	011 : 48,8	010 : 56,0	000 : 56,0	001 : 55,0
111 : 48,8	110 : 58,5	000 : 51,5	111 : 58,5	110 : 59,0	111 : 53,8

Tabla 10.2. Arreglo de tratamientos para la pureza observada de un producto químico en un factorial  $2^3$  confundido totalmente.

En este experimento, se tienen dos pasos en el proceso de la aleatorización:

a) Decidir qué bloques llevan  $(ABC)_0$  y qué bloques llevan  $(ABC)_1$ , y

b) Decidir la aleatorización de los tratamientos dentro de cada bloque.

Con los datos presentados en la tabla se realiza el análisis de varianza usual. El efecto de tratamientos se descompone en seis contrastes ortogonales que representan los efectos principales y las interacciones, como se muestra.

## Solución

```
#####  
## Factorial 2^3 BLOQUEADO (ABC confundido)  
## 3 réplicas, 2 bloques por réplica (6 bloques)  
##  
#####  
  
## 1) Paquetes ----  
## FrF2/DoE.base dan utilidades para diseños factoriales; broom deja "tablas"  
## con buen formato  
  
# install.packages(c("FrF2", "DoE.base", "broom"))  
  
library(FrF2)  
  
library(DoE.base)  
library(broom)  
  
## 2) Ingresamos Los datos EXACTOS de la tabla ----  
## Cada fila es una corrida: niveles de A,B,C en 0/1 y la respuesta "Y"  
## (pureza).  
## También indicamos Réplica (I, II, III) y Bloque (1..6) como en la figura.  
  
dat <- rbind(  
  ## Réplica I -----  
  ## Bloque 1: (ABC)_1  
  data.frame(replica="I", bloque=1, A=0, B=0, C=1, Y=44.5), # 001  
  data.frame(replica="I", bloque=1, A=0, B=1, C=0, Y=44.2), # 010  
  data.frame(replica="I", bloque=1, A=1, B=0, C=0, Y=60.1), # 100  
  data.frame(replica="I", bloque=1, A=1, B=1, C=1, Y=48.8), # 111  
  
  ## Bloque 2: (ABC)_0  
  data.frame(replica="I", bloque=2, A=0, B=0, C=0, Y=46.8), # 000  
  data.frame(replica="I", bloque=2, A=0, B=1, C=1, Y=44.5), # 011  
  data.frame(replica="I", bloque=2, A=1, B=0, C=1, Y=57.0), # 101  
  data.frame(replica="I", bloque=2, A=1, B=1, C=0, Y=58.5), # 110  
  
  ## Réplica II -----  
  ## Bloque 3: (ABC)_0  
  data.frame(replica="II", bloque=3, A=1, B=0, C=1, Y=49.8), # 101  
  data.frame(replica="II", bloque=3, A=1, B=1, C=0, Y=52.0), # 110  
  data.frame(replica="II", bloque=3, A=0, B=1, C=1, Y=48.8), # 011
```

```

data.frame(replica="II", bloque=3, A=0, B=0, C=0, Y=51.5), # 000

## Bloque 4: (ABC)_1
data.frame(replica="II", bloque=4, A=0, B=0, C=1, Y=55.5), # 001
data.frame(replica="II", bloque=4, A=1, B=0, C=0, Y=59.8), # 100
data.frame(replica="II", bloque=4, A=0, B=1, C=0, Y=56.0), # 010
data.frame(replica="II", bloque=4, A=1, B=1, C=1, Y=58.5), # 111

## Réplica III -----
## Bloque 5: (ABC)_0
data.frame(replica="III", bloque=5, A=0, B=1, C=1, Y=53.2), # 011
data.frame(replica="III", bloque=5, A=1, B=0, C=1, Y=57.2), # 101
data.frame(replica="III", bloque=5, A=0, B=0, C=0, Y=56.0), # 000
data.frame(replica="III", bloque=5, A=1, B=1, C=0, Y=59.0), # 110

## Bloque 6: (ABC)_1
data.frame(replica="III", bloque=6, A=1, B=0, C=0, Y=69.5), # 100
data.frame(replica="III", bloque=6, A=0, B=1, C=0, Y=62.8), # 010
data.frame(replica="III", bloque=6, A=0, B=0, C=1, Y=55.0), # 001
data.frame(replica="III", bloque=6, A=1, B=1, C=1, Y=53.8) # 111
)

## Chequeo rápido de tamaños (deben ser 24 corridas = 8 tratamientos * 3
réplicas)
nrow(dat)

## [1] 24

table(dat$replica)

##
##  I  II III
##  8   8   8

table(dat$bloque)

##
## 1 2 3 4 5 6
## 4 4 4 4 4 4

## 3) Creamos variables útiles para DOE ----
## - Factores en codificación -1/+1 (estándar para calcular efectos)
## - Factor "Blocks" (6 niveles) y "Replica" (3 niveles)
dat$Apm <- ifelse(dat$A==0, -1, 1) # A plus-minus
dat$Bpm <- ifelse(dat$B==0, -1, 1)
dat$Cpm <- ifelse(dat$C==0, -1, 1)

dat$Blocks <- factor(dat$bloque) # factor de 6 niveles
dat$Replica <- factor(dat$replica, levels=c("I","II","III"))

```

```
## (Opcional didáctico) Verificamos la regla de bloqueo: (ABC)_0 vs (ABC)_1
## En 0/1: bloque tipo 0 si (A+B+C) %% 2 == 0; tipo 1 si == 1
dat$ABCparidad <- (dat$A + dat$B + dat$C) %% 2 # 0 -> (ABC)_0 ; 1 -> (ABC)_1
with(dat, tapply(ABCparidad, Blocks, unique)) # inspecciona qué paridad
tiene cada bloque
```

```
## 1 2 3 4 5 6
## 1 0 0 1 0 1
```

## 4) Ajustamos el modelo ANOVA con bloques ----

## Importante: al confundir ABC con bloques, el efecto ABC se "va" al término Blocks.

## Por eso, en el modelo sólo incluimos Principales + Dos-factores (A, B, C, AB, AC, BC)

## y añadimos "Blocks" como efecto de bloqueo (factor fijo aquí).

```
fit <- aov(
  Y ~ Blocks + (Apm + Bpm + Cpm)^2, # principales e interacciones de 2
  data = dat)
factores
```

## Resumen ANOVA

```
summary(fit)
```

```
##              Df Sum Sq Mean Sq F value    Pr(>F)
## Blocks         5  379.4    75.88    5.121 0.00960 **
## Apm            1  177.1   177.13   11.955 0.00474 **
## Bpm            1   21.3    21.28    1.436 0.25386
## Cpm            1  102.5   102.51    6.918 0.02196 *
## Apm:Bpm        1   22.0    22.04    1.488 0.24601
## Apm:Cpm        1   13.5    13.50    0.911 0.35864
## Bpm:Cpm        1    0.0     0.00    0.000 0.99171
## Residuals     12  177.8    14.82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## 5) Tabla ordenada (broom) y estimación de efectos ----

```
tidy_fit <- broom::tidy(fit)
tidy_fit[order(tidy_fit$term), ]
```

```
## # A tibble: 8 × 6
##   term          df      sumsq    meansq statistic  p.value
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 Apm          1  177.      177.      12.0      0.00474
## 2 Apm:Bpm      1   22.0      22.0       1.49      0.246
## 3 Apm:Cpm      1   13.5      13.5       0.911     0.359
## 4 Blocks       5  379.      75.9       5.12      0.00960
## 5 Bpm          1   21.3      21.3       1.44      0.254
## 6 Bpm:Cpm      1   0.00167    0.00167    0.000112   0.992
## 7 Cpm          1  103.      103.       6.92      0.0220
## 8 Residuals   12  178.      14.8       NA         NA
```

```
## Efectos estimados a partir de un modelo lineal con codificación -1/+1
## (esto te permite ver magnitudes y signos de efectos)
```

```
fit_lm <- lm(Y ~ Blocks + (Apm + Bpm + Cpm)^2, data = dat)
coefs <- coef(summary(fit_lm))
coefs
```

```
##              Estimate Std. Error    t value    Pr(>|t|)
## (Intercept) 49.400000000  1.9246031 25.66762941 7.448891e-12
## Blocks2      2.300000000  2.7217999  0.84502907 4.146256e-01
## Blocks3      1.125000000  2.7217999  0.41332944 6.866557e-01
## Blocks4      8.050000000  2.7217999  2.95760175 1.197389e-02
## Blocks5      6.950000000  2.7217999  2.55345741 2.530428e-02
## Blocks6     10.875000000  2.7217999  3.99551789 1.775964e-03
## Apm          2.716666667  0.7857159  3.45756848 4.738358e-03
## Bpm         -0.941666667  0.7857159 -1.19848233 2.538630e-01
## Cpm         -2.066666667  0.7857159 -2.63029750 2.196361e-02
## Apm:Bpm     -0.958333333  0.7857159 -1.21969440 2.460110e-01
## Apm:Cpm     -0.750000000  0.7857159 -0.95454345 3.586427e-01
## Bpm:Cpm     -0.008333333  0.7857159 -0.01060604 9.917121e-01
```

```
## 6) ¿Qué pasa con ABC? (alias con bloques) ----
## No incluimos ABC en el modelo porque está "no estimable" al estar
## perfectamente
## confundido con Blocks: cualquier cambio sistemático de ABC es absorbido
## por Blocks.
## Para visualizar aliasing teórico usamos un diseño equivalente con FrF2:
```

```
## Generamos el diseño 2^3 con 3 repeticiones, 2 bloques por réplica,
## bloqueando con ABC
## (sin respuesta; esto es sólo para imprimir la estructura de alias)
des <- FrF2(nruns      = 8,
            nfactors   = 3,
            blocks      = 2,          # 2 bloques por réplica
            replications = 3,        # 3 réplicas -> 6 bloques
            block.gen    = "ABC",     # interacción que define bloques
            randomize    = FALSE)     # didáctico: sin aleatorizar
```

```
## Estructura de alias (respecto a principales e interacciones de 2 factores)
aliasprint(des) # verás que ABC aparece en la columna de bloques
```

```
## $legend
## [1] A=A B=B C=C
```

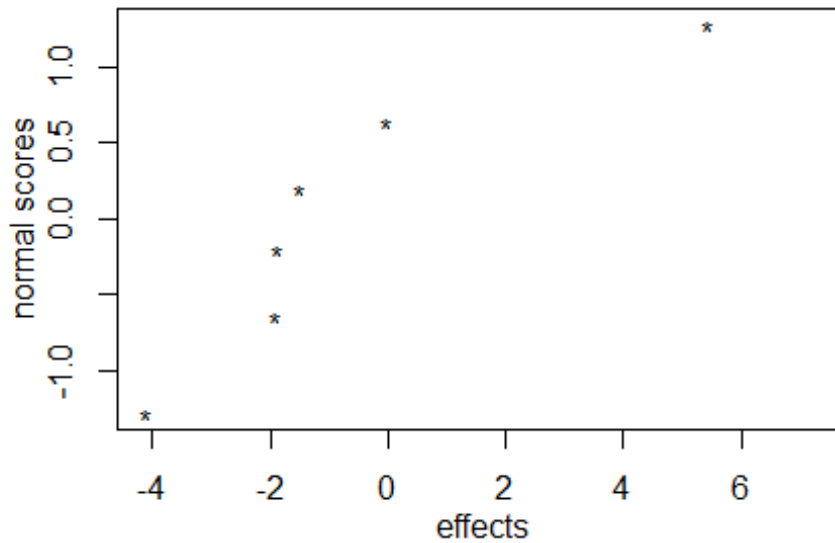
```
## 7) Gráficos de diagnóstico y de efectos ----
## (a) Gráfico de Daniel (normal de efectos) para detectar efectos "grandes"
## Nota: DanielPlot necesita un objeto aov con sólo los términos de
## tratamiento.
## Le quitamos Blocks para ese gráfico específico:
```

```
fit_trt <- aov(Y ~ (Apm + Bpm + Cpm)^2, data = dat) # sólo tratamientos
```

```
par(mfrow=c(1,1))
DanielPlot(fit_trt, main="Daniel Plot - Efectos de tratamiento (sin Blocks)")

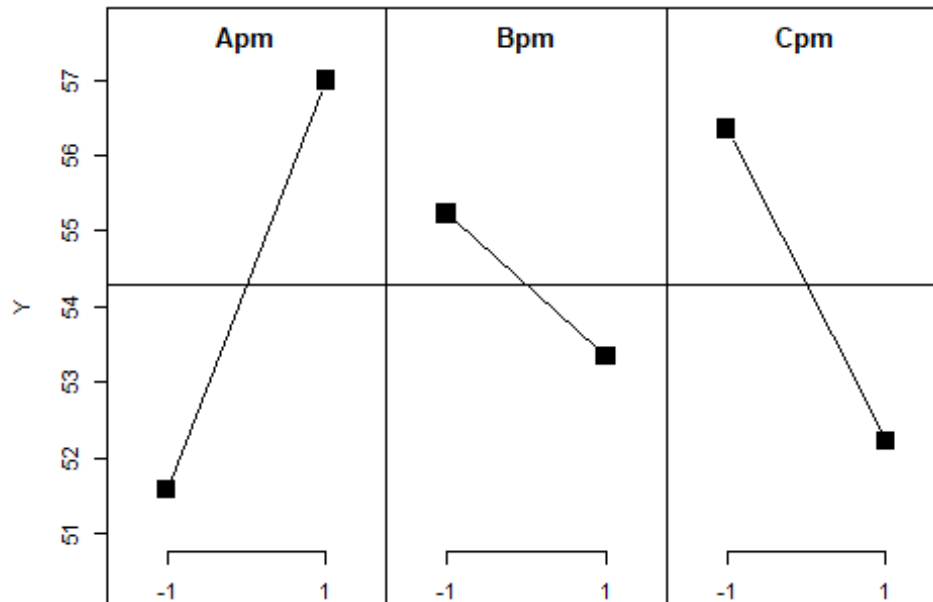
## simulated critical values not available for all requests, used
conservative ones
```

### Daniel Plot - Efectos de tratamiento (sin Blocks)



```
## (b) Efectos principales
MEPlot(fit_trt, main="Efectos principales")
```

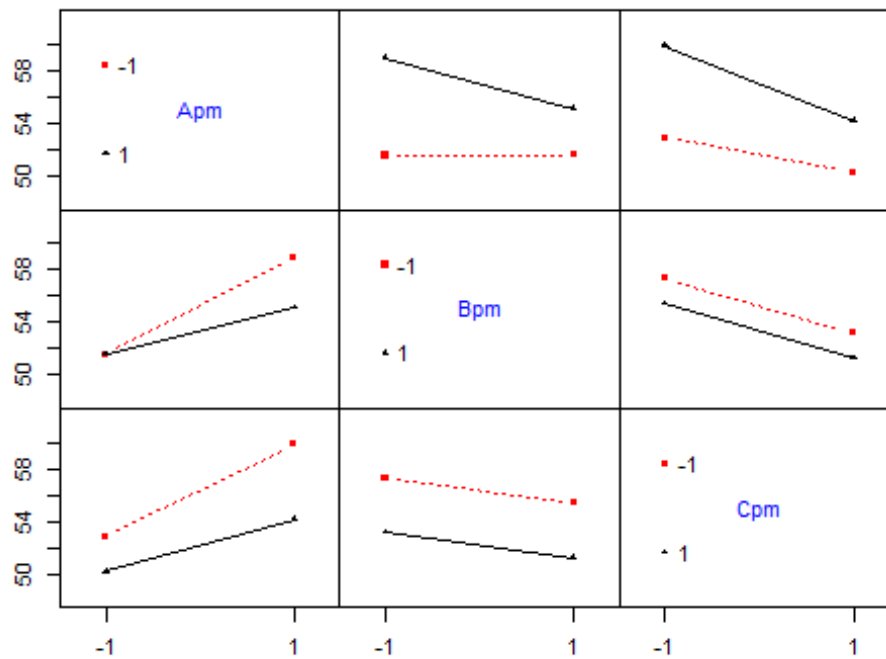
## Efectos principales



## (c) Interacciones de dos factores

IAPlot(fit\_trt, main="Interacciones de dos factores")

## Interacciones de dos factores



```
## 8) Medias por tratamiento y por bloque (resumen) --
## Tratamiento = combinación -1/+1 de A,B,C
dat$trt <- with(dat, interaction(Apm,Bpm,Cpm, sep=""))
aggregate(Y ~ trt, data = dat, FUN = mean) # medias por tratamiento

##      trt      Y
## 1 -1-1-1 51.43333
## 2  1-1-1 63.13333
## 3 -11-1 54.33333
## 4  11-1 56.50000
## 5 -1-11 51.66667
## 6  1-11 54.66667
## 7 -111 48.83333
## 8  111 53.70000

aggregate(Y ~ Blocks, data = dat, FUN = mean) # medias por bloque (días)

##  Blocks      Y
## 1      1 49.400
## 2      2 51.700
## 3      3 50.525
## 4      4 57.450
## 5      5 56.350
## 6      6 60.275

## 9) Breve interpretación guiada (para el informe) --
## - EL término Blocks captura variación entre días.
## Como ABC está confundido con Blocks,
## no podemos separar "efecto ABC" del "efecto día". Por eso ABC NO se
estima.
## - Observa en summary(fit) qué principales/interacciones
## 2-factores resultan significativas.
## - Usa MEPlot/IAPlot para visualizar qué factores elevan la
pureza y cómo interactúan.
```