

Metodología Six Sigma

Herramientas para la fase de mejora: Diseño de experimentos con R

Profesor: Gustavo Ahumada

Niveles de factores y replicas

Ejemplo (Masa para pizza). Un fabricante de alimentos está buscando la mejor receta para su producto principal: venta de masa para pizza al por menor a través de su área de mercado. Para descubrir *la formula magica*, el manager decide realizar un experimento para determinar los niveles óptimos de los tres ingredientes principales en la masa de pizza: harina, sal y levadura en polvo. Los otros ingredientes son fijos ya que no afectan el sabor de la pizza cocida final. El panel de expertos determinará el sabor del producto y le dará una puntuación a cada receta. Por lo tanto, tenemos tres factores que llamaremos harina, sal y bakPow, con dos niveles cada uno (– y +). Luego podemos construir una hoja de datos para registrar los datos medidos usando el siguiente código:

```
Diseno_pizza <- expand.grid(flour = gl(2, 1, labels = c("-",
"+")),
sal = gl(2, 1, labels = c("-", "+")),
polvo_hornear = gl(2, 1, labels = c("-", "+")),
calificacion = NA)
Diseno_pizza
```

##	flour	sal	polvo_hornear	calificacion
## 1	-	-	-	NA
## 2	+	-	-	NA
## 3	-	+	-	NA
## 4	+	+	-	NA
## 5	-	-	+	NA
## 6	+	-	+	NA
## 7	-	+	+	NA
## 8	+	+	+	NA

Supuestos del modelo

Recuerde que estamos en la fase de Mejora del ciclo DMAIC, y hemos llegado aquí después de haber definido el proceso y los objetivos del proyecto.

Para validar los resultados, debemos reiterar la importancia de la aleatorización. Una vez que haya corregido los factores y los niveles de un experimento, las mediciones deben tomarse al azar. Por lo tanto, si tenemos un pedido. En la lista de todas las combinaciones de niveles de factores, antes de comenzar a medir la respuesta, debemos establecer el orden aleatorio en el que se van a medir.

La técnica estadística adecuada para analizar experimentos es el análisis de varianza (ANOVA). Existen dos supuestos principales en este tipo de modelo: (1) la normalidad e independencia de los residuos y (2) la homocedasticidad (varianza constante). Una estrategia sistemática para la aleatorización es útil para lograr la primera, mientras que la segunda puede verificarse mediante pruebas de hipótesis.

Ejemplo (Masa para pizza (continuación)). Una vez que se ha diseñado un experimento, procederemos con su aleatorización. Podemos agregar una columna a la matriz de diseño con el orden aleatorio de los ensayos y ordenar las filas de la matriz de diseño siguiendo un orden aleatorio:

```
Diseno_pizza$ord <- sample(1:8, 8)
Diseno_pizza[order(Diseno_pizza$ord),]
```

```
##   flour sal polvo_hornear calificacion ord
## 2    +   -              -            NA   1
## 1    -   -              -            NA   2
## 4    +   +              -            NA   3
## 7    -   +              +            NA   4
## 5    -   -              +            NA   5
## 3    -   +              -            NA   6
## 6    +   -              +            NA   7
## 8    +   +              +            NA   8
```

Diseños factoriales 2^k

Diseños factoriales 2^k son aquellos donde el número de factores para ser estudiados son k , cada uno de ellos con 2 niveles. El número total de experimentos que se llevarán a cabo para obtener una completa replica es la potencia 2^k (2 a la k). Si se quiere n replicas del experimento, entonces el número total de experimentos es $n * 2^k$.

Usando ANOVA, calcularemos el efecto de cada factor e interacción y evaluaremos cuáles de estos efectos son significativos. Representemos los factores con letras mayúsculas latinas (A, B, ...). Los efectos principales generalmente están representados por letras griegas (α, β, \dots) Correspondientes a la letra latina del factor, y los efectos de las interacciones por la combinación de las letras que representan los factores cuyos efectos interactúan. Por ejemplo, para un experimento 2^k con tres factores y n repeticiones, el modelo estadístico es:

$$y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + (\alpha\beta)_{ij} + (\alpha\gamma)_{ik} + (\beta\gamma)_{jk} + (\alpha\beta\gamma)_{ijk} + \varepsilon_{ijkl} \quad (1)$$

$$i = 1, 2; \quad j = 1, 2; \quad k = 1, 2; \quad l = 1, \dots, n$$

$$\varepsilon_{ijkl} \sim N(0, \sigma) \quad \text{independiente}$$

donde μ es la media global de la respuesta,

α_i es el efecto del factor A en el nivel i ;

β_j es el efecto del factor B en el nivel j ;

γ_k es el efecto del factor C en el nivel k ;

$(\alpha\beta)_{ij}$ es el efecto de la interacción entre el factor A y B en el nivel i y j , respectivamente.

$(\alpha\gamma)_{ik}$ es el efecto de la interacción entre el factor A y C en el nivel i y k , respectivamente;

$(\beta\gamma)_{jk}$ es el efecto de la interacción entre el factor B y C en el nivel j y k , respectivamente;

$(\alpha\beta\gamma)_{ijk}$ es el efecto de la interacción de los factores A, B y C en los niveles i, j y k , respectivamente.

Ejemplo (Masa para pizza (continuación)). El experimento se lleva a cabo preparando las pizzas en la fábrica siguiendo las instrucciones del paquete, que habían sido probadas previamente como las mejores condiciones para preparar la pizza, a saber: "hornee la pizza durante 9 minutos en un horno a 180 grados celsius.

Después de una votación a ciegas, las evaluaciones dadas por los expertos para cada uno de los ocho experimentos (2^3) en cada una de las replicas.

```
library(SixSigma)
ss.data.doe1
```

```
##      repl flour salt bakPow score ord
## 1      1      -      -      -  5.33  2
## 2      1      +      -      -  6.99  4
## 3      1      -      +      -  4.23  8
## 4      1      +      +      -  6.61  5
## 5      1      -      -      +  2.26  1
## 6      1      +      -      +  5.75  6
## 7      1      -      +      +  3.26  3
## 8      1      +      +      +  6.24  7
## 9      2      -      -      -  5.70  2
## 10     2      +      -      -  7.71  4
## 11     2      -      +      -  5.13  8
## 12     2      +      +      -  6.76  5
## 13     2      -      -      +  2.79  1
## 14     2      +      -      +  4.57  6
## 15     2      -      +      +  2.48  3
## 16     2      +      +      +  6.18  7
```

```
aggregate(score ~ flour + salt + bakPow,
           FUN = mean, data = ss.data.doe1)
```

```
##      flour salt bakPow score
## 1      -      -      -  5.515
## 2      +      -      -  7.350
## 3      -      +      -  4.680
## 4      +      +      -  6.685
## 5      -      -      +  2.525
## 6      +      -      +  5.160
## 7      -      +      +  2.870
## 8      +      +      +  6.210
```

```
doe.model1 <- lm(score ~ flour + salt + bakPow +
  flour * salt + flour * bakPow +
  salt * bakPow + flour * salt * bakPow,
  data = ss.data.doe1)
summary(doe.model1)
```

```
##
## Call:
## lm(formula = score ~ flour + salt + bakPow + flour * salt + flour *
##      bakPow + salt * bakPow + flour * salt * bakPow, data = ss.data.doe1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.5900 -0.2888  0.0000  0.2888  0.5900
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.5150     0.3434  16.061 2.27e-07 ***
## flour+           1.8350     0.4856   3.779 0.005398 **
## salt+           -0.8350     0.4856  -1.719 0.123843
## bakPow+          -2.9900     0.4856  -6.157 0.000272 ***
## flour+:salt+      0.1700     0.6868   0.248 0.810725
## flour+:bakPow+    0.8000     0.6868   1.165 0.277620
## salt+:bakPow+     1.1800     0.6868   1.718 0.124081
## flour+:salt+:bakPow+ 0.5350     0.9712   0.551 0.596779
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4856 on 8 degrees of freedom
## Multiple R-squared:  0.9565, Adjusted R-squared:  0.9185
## F-statistic: 25.15 on 7 and 8 DF,  p-value: 7.666e-05
```

```
doe.model2 <- lm(score ~ flour + bakPow,
data = ss.data.doe1)
summary(doe.model2)
```

```
##
## Call:
## lm(formula = score ~ flour + bakPow, data = ss.data.doe1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84812 -0.54344  0.06062  0.44406  0.86938
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.8306     0.2787  17.330 2.30e-10 ***
## flour+         2.4538     0.3219   7.624 3.78e-06 ***
## bakPow+        -1.8662     0.3219  -5.798 6.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6437 on 13 degrees of freedom
## Multiple R-squared:  0.8759, Adjusted R-squared:  0.8568
## F-statistic: 45.87 on 2 and 13 DF,  p-value: 1.288e-06
```

```
coef(doe.model2)
```

```
## (Intercept)      flour+      bakPow+
##    4.830625    2.453750   -1.866250
```

Por lo tanto, el modelo estadístico para el experimento es

$$\widehat{score} = 4.830625 + 2.453750 \times flour - 1.866250 \times bakPow, \quad (2)$$

expresado en términos de los coeficientes del modelo. En la ecuación precedente, los posibles valores para *flour* y *bakPow* son 0 para un bajo nivel (−) y 1 para el mayor nivel (+).

Esta codificación de los factores es diferente de la ecuación (1), en la cual los posibles valores para los factores son -1 para el menor nivel y $+1$ para el mayor nivel. Para calcular los efectos, los coeficientes deberían ser divididos por dos y el modelo equivalente a (1) sería

$$\widehat{score} = 5.1244 + 1.2269 \times flour - 0.9331 \times backPow, \quad (3)$$

donde 5.1244 es la evaluación promedio. Por ejemplo, en esta formulación del modelo, pasando *flour* del nivel -1 al nivel $+1$ implica un incremento de 2.453750 en la evaluación promedio de la receta.

Si queremos cambiar del modelo dado por (3) al modelo dado por (2), entonces todo lo que tenemos que hacer es calcular los coeficientes duplicando los efectos y calculando la intersección como:

$$\bar{y} - \alpha_1 - \beta_1 = 5.1244 - 1.2269 - (-0.9331) = 4.830625 \quad (4)$$

```
as.numeric(mean(ss.data.doe1$score) - coef(doe.model2)[2]/2 - coef(doe.model2)[3]/2)
```

```
## [1] 4.830625
```

Por lo tanto, la receta con un alto nivel de *flour* y un bajo nivel de *baking powder* será la mejor, independientemente del nivel de *salt* (alta o baja). El puntaje estimado para esta receta es $4.8306 + 2.4538 \times 1 + (-1.8662) \times 0 = 7.284$:

```
coef(doe.model2)[1] + coef(doe.model2)[2]
```

```
## (Intercept)
##      7.284375
```

Obtenemos resultados similares empleando la fórmula del efecto:

$$\hat{y} = 5.1244 + 1.2269 \times 1 + (-0.9331) \times (-1) = 7.2844. \quad (5)$$

```
mean(ss.data.doe1$score) + coef(doe.model2)[2]/2*(1) + coef(doe.model2)[3]/2*(-1)
```

```
## flour+
## 7.284375
```

Se puede obtener las estimaciones para todas las condiciones experimentales (incluyendo las replicas) empleando la función *predict*

```
predict(doe.model2)
```

```
##      1      2      3      4      5      6      7      8
## 4.830625 7.284375 4.830625 7.284375 2.964375 5.418125 2.964375 5.418125
##      9     10     11     12     13     14     15     16
## 4.830625 7.284375 4.830625 7.284375 2.964375 5.418125 2.964375 5.418125
```

La estimación para las condiciones experimentales 2, 4, 10 y 12 son las mismas (7.2844) que el factor de sal se ha eliminado del modelo.

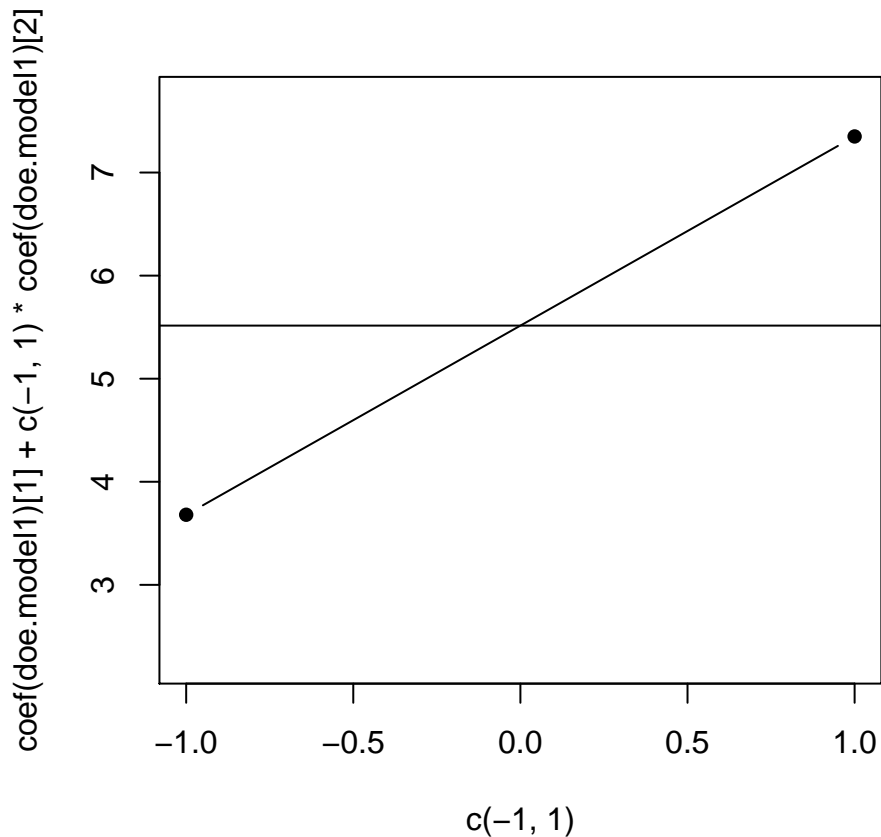
También podemos calcular un intervalo de confianza para cada parámetro

```
confint(doe.model2)
```

```
##              2.5 %    97.5 %  
## (Intercept)  4.228435  5.432815  
## flour+      1.758401  3.149099  
## bakPow+     -2.561599 -1.170901
```

Para visualizar gráficamente los efectos principales, se puede usar la función *plot*. En este caso, se trazarán los puntos y líneas que muestran los efectos. Por ejemplo, para representar los efectos del factor *flour* (ver figura debajo), escribir

```
plot(c(-1, 1), ylim = range(ss.data.doe1$score),  
coef(doe.model1)[1] + c(-1, 1) * coef(doe.model1)[2],  
type="b", pch=16)  
abline(h=coef(doe.model1)[1])
```



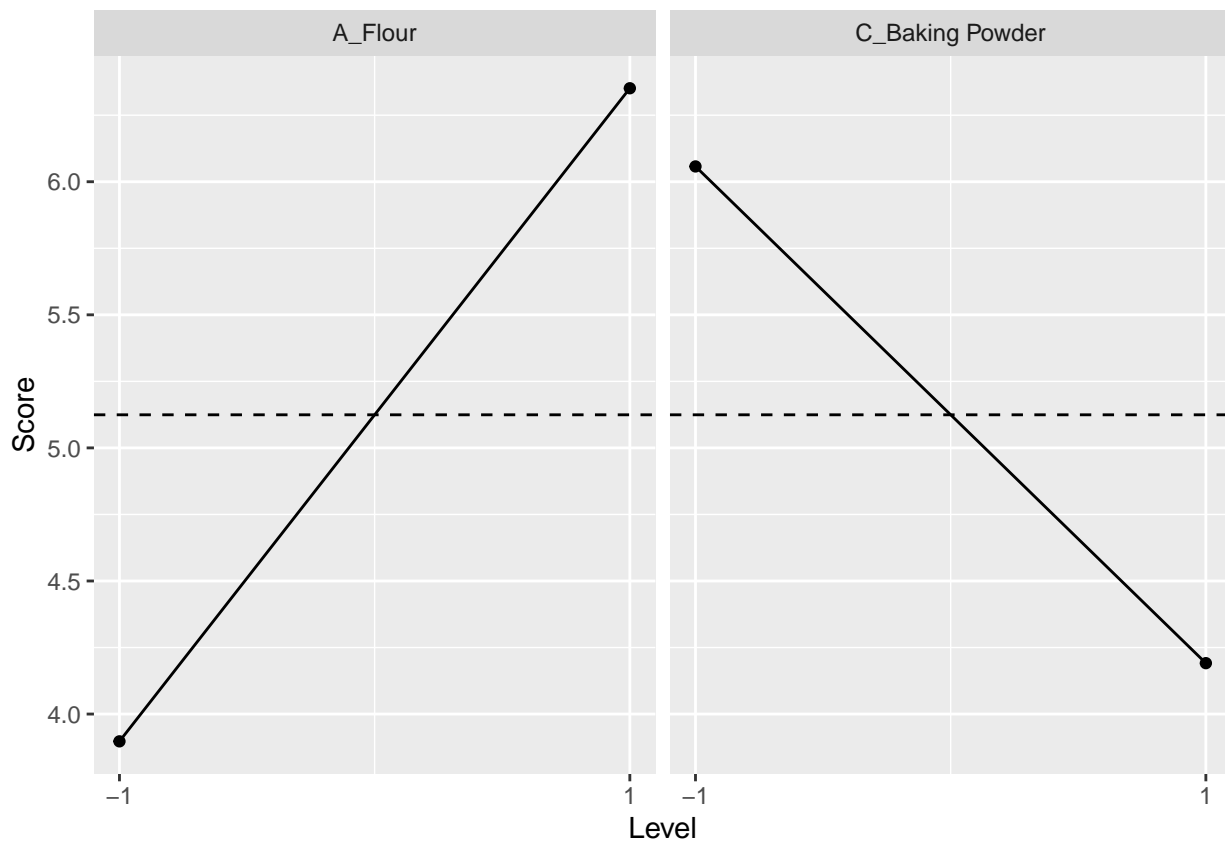
También podemos trazar gráficos para visualizar los efectos con el paquete *ggplot2*. Usando el siguiente código graficamos los efectos principales de los dos factores en el mismo gráfico (ver figura debajo):

```
library(ggplot2)  
prinEf <- data.frame(Factor = rep(c("A_Flour",  
"C_Baking Powder"), each = 2),
```

```

Level = rep(c(-1, 1), 2),
Score = c(aggregate(score ~ flour, FUN = mean, data =
ss.data.doe1)[,2],
aggregate(score ~ bakPow, FUN = mean, data =
ss.data.doe1)[,2]))
p <- ggplot(prinEf,
aes(x = Level, y = Score)) +
geom_point() +
geom_line() +
scale_x_continuous(breaks = c(-1, 1)) +
facet_grid(. ~ Factor) +
geom_abline(intercept = mean(ss.data.doe1$score),
slope = 0, linetype = "dashed") +
theme(plot.background = element_rect(colour="white"))
print(p)

```

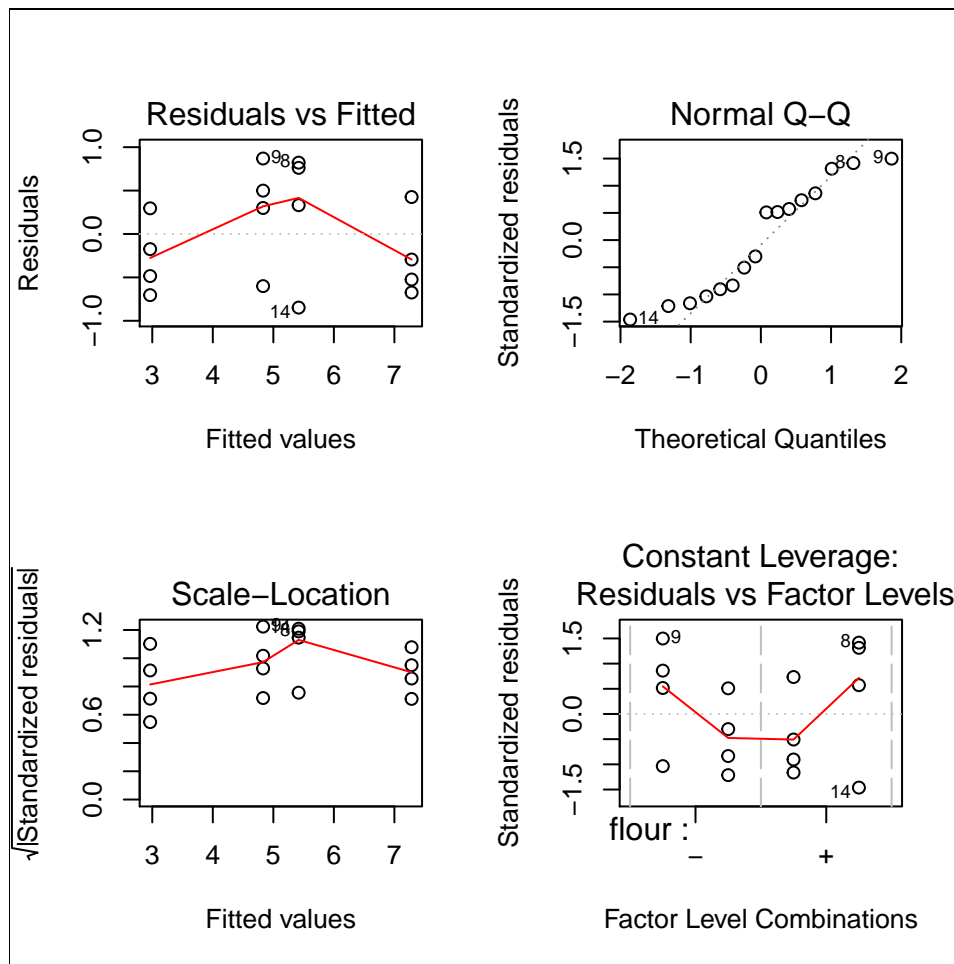


El diagnóstico del modelo debe hacerse analizando los residuos. Podemos trazar los gráficos estándar para modelos lineales (ver gráfico debajo):

```

par(mfrow=c(2,2))
plot(doe.model12)
box("outer")

```



```
shapiro.test(residuals(doe.model2))
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(doe.model2)
## W = 0.90652, p-value = 0.1023
```

Diseño de experimentos para la mejora de procesos

Hemos descrito un Diseño de experimento completo, pero realizar experimentos puede no ser suficiente para mejorar un proceso. Por ejemplo, quizás no se han identificado todas las X del proceso, o algunas X pueden depender de condiciones externas y, por lo tanto, no están bajo nuestro control. En estas circunstancias, una posibilidad es diseñar productos que sean robustos para el medio ambiente (diseño robusto). Esto consiste en incluir factores de ruido dentro del experimento. Estos factores de ruido son aquellos que pueden afectar las características de CTQ, pero no están bajo nuestro control. Debemos identificar todos los posibles factores de ruido. Podemos obtener información valiosa sobre estas condiciones externas del propietario del proceso.

Ejemplo (Masa para pizza (continuación)). Después de unas semanas, el administrador de redes sociales del departamento de marketing informó a la junta directiva que se habían recogido algunos malos comentarios de las redes sociales (principalmente Facebook y Twitter) con respecto a una mala experiencia

que involucraba a los consumidores de la masa de pizza. El sabor de las pizzas hechas con la masa no fue satisfactorio.

Para investigar las razones, decidieron abordar el problema desde una perspectiva **Six Sigma**. Durante la fase de medición, se realizó una identificación precisa de los posibles factores que afectan el sabor de la pizza. El propietario del proceso proporcionó una información vital (el miembro del equipo que presentó las instrucciones de preparación). Señaló que las condiciones exactas de horneado (temperatura = 180 °C y tiempo = 9 minutos) fueron críticas para lograr el sabor deseado por los clientes.

Investigaciones adicionales sobre cómo los clientes preparaban las pizzas mostraron que no todos (en realidad, casi nadie) seguían con precisión las instrucciones de preparación en la caja. Por lo tanto, el *Black Belt* propuso un nuevo diseño para realizar una nueva investigación sobre la receta de masa de pizza. Decidió incluir dos factores de ruido con dos niveles cada uno: tiempo (7 y 11 min, $-t$ y $+t$, respectivamente) y temperatura (160°C y 200°C, $-T$ y $+T$, respectivamente). Ahora tenemos un diseño factorial de 2^5 , con la siguiente matriz de diseño:

```
Diseno_pizza2 <- expand.grid(flour = gl(2, 1, labels = c("-", "+")),
  salt = gl(2, 1, labels = c("-", "+")),
  bakPow = gl(2, 1, labels = c("-", "+")),
  temp = gl(2, 1, labels = c("-", "+")),
  time = gl(2, 1, labels = c("-", "+")),
  score = NA)
Diseno_pizza2
```

##	flour	salt	bakPow	temp	time	score
## 1	-	-	-	-	-	NA
## 2	+	-	-	-	-	NA
## 3	-	+	-	-	-	NA
## 4	+	+	-	-	-	NA
## 5	-	-	+	-	-	NA
## 6	+	-	+	-	-	NA
## 7	-	+	+	-	-	NA
## 8	+	+	+	-	-	NA
## 9	-	-	-	+	-	NA
## 10	+	-	-	+	-	NA
## 11	-	+	-	+	-	NA
## 12	+	+	-	+	-	NA
## 13	-	-	+	+	-	NA
## 14	+	-	+	+	-	NA
## 15	-	+	+	+	-	NA
## 16	+	+	+	+	-	NA
## 17	-	-	-	-	+	NA
## 18	+	-	-	-	+	NA
## 19	-	+	-	-	+	NA
## 20	+	+	-	-	+	NA
## 21	-	-	+	-	+	NA
## 22	+	-	+	-	+	NA
## 23	-	+	+	-	+	NA
## 24	+	+	+	-	+	NA
## 25	-	-	-	+	+	NA
## 26	+	-	-	+	+	NA
## 27	-	+	-	+	+	NA
## 28	+	+	-	+	+	NA
## 29	-	-	+	+	+	NA
## 30	+	-	+	+	+	NA

```
## 31      -      +      +      +      +      NA
## 32      +      +      +      +      +      NA
```

```
library(SixSigma)
ss.data.doe2
```

```
##      repl flour salt bakPow temp time score
## 1      1      -      -      -      -      - 3.55
## 2      1      +      -      -      -      - 3.13
## 3      1      -      +      -      -      - 0.66
## 4      1      +      +      -      -      - 3.54
## 5      1      -      -      +      -      - 1.74
## 6      1      +      -      +      -      - 3.53
## 7      1      -      +      +      -      - 1.47
## 8      1      +      +      +      -      - 5.83
## 9      1      -      -      -      +      - 5.17
## 10     1      +      -      -      +      - 4.38
## 11     1      -      +      -      +      - 4.96
## 12     1      +      +      -      +      - 6.05
## 13     1      -      -      +      +      - 3.21
## 14     1      +      -      +      +      - 5.07
## 15     1      -      +      +      +      - 1.89
## 16     1      +      +      +      +      - 6.25
## 17     1      -      -      -      -      + 4.39
## 18     1      +      -      -      -      + 5.05
## 19     1      -      +      -      -      + 5.49
## 20     1      +      +      -      -      + 5.44
## 21     1      -      -      +      -      + 1.85
## 22     1      +      -      +      -      + 5.30
## 23     1      -      +      +      -      + 3.20
## 24     1      +      +      +      -      + 5.94
## 25     1      -      -      -      +      + 3.93
## 26     1      +      -      -      +      + 2.84
## 27     1      -      +      -      +      + 1.16
## 28     1      +      +      -      +      + 3.90
## 29     1      -      -      +      +      + 1.75
## 30     1      +      -      +      +      + 2.84
## 31     1      -      +      +      +      + 1.30
## 32     1      +      +      +      +      + 5.64
## 33     2      -      -      -      -      - 3.80
## 34     2      +      -      -      -      - 3.61
## 35     2      -      +      -      -      - 1.25
## 36     2      +      +      -      -      - 3.64
## 37     2      -      -      +      -      - 2.09
## 38     2      +      -      +      -      - 2.75
## 39     2      -      +      +      -      - 0.96
## 40     2      +      +      +      -      - 5.78
## 41     2      -      -      -      +      - 5.07
## 42     2      +      -      -      +      - 4.66
## 43     2      -      +      -      +      - 4.86
## 44     2      +      +      -      +      - 5.74
## 45     2      -      -      +      +      - 2.90
## 46     2      +      -      +      +      - 4.95
## 47     2      -      +      +      +      - 1.83
```

## 48	2	+	+	+	+	-	5.97
## 49	2	-	-	-	-	+	3.98
## 50	2	+	-	-	-	+	5.05
## 51	2	-	+	-	-	+	5.10
## 52	2	+	+	-	-	+	5.81
## 53	2	-	-	+	-	+	1.60
## 54	2	+	-	+	-	+	5.77
## 55	2	-	+	+	-	+	2.88
## 56	2	+	+	+	-	+	6.02
## 57	2	-	-	-	+	+	3.87
## 58	2	+	-	-	+	+	3.04
## 59	2	-	+	-	+	+	1.18
## 60	2	+	+	-	+	+	3.84
## 61	2	-	-	+	+	+	1.65
## 62	2	+	-	+	+	+	2.96
## 63	2	-	+	+	+	+	1.32
## 64	2	+	+	+	+	+	6.29