Desarrollo de modelos matemáticos

Juan David Argüello Plata

Universidad Industrial de Santander

En el presente trabajo, se desarrolla un análisis referente al planteamiento de modelos matemáticos a partir de datos experimentales obtenidos de un diseño experimental fraccionado - NIST.

Nos centramos tanto en el plantemiento del modelo como en la validación del mismo; pero antes, es necesario hablar un poco sobre los datos experimentales.

Datos experimentales

Los datos experimentales definen tanto el número de variables que compondrán el modelo como de la cantidad de modelos matemáticos posibles.

Empezamos importando y analizando el modelo...

In [ ]:

```
from App.DirTree import list_files
import os

current_dir = os.getcwd()
list_files(current_dir)
```

In [1]:

```
archivo = '1'
path_file = 'App/DataBase/Datos/' + archivo + '.txt'
with open(path_file, encoding='latin-1') as file:
    info = file.read()
print(info)
%store archivo
```

```
A       B       C       Y
-1      -1      -1      6,073
1       -1      -1      2,447
-1      1       -1      1,559
1       1       -1      5,745
-1      -1      1       7,799
1       -1      1       3,667
-1      1       1       3,863
1       1       1       8,201
0       0       2       5,777
0       0       -2      0,832
0       2       0       3,267
0       -2      0       2,705
2       0       0       14,853
-2      0       0       9,350
0       1,5     0       3,129
0       -1,5    0       2,973
1,5     0       0       8,642
-1,5    0       0       6,511
0       0       1,5     4,689
0       0       -1,5    1,418
1       1       0       9,135
-1      -1      0       0,016
0       0       0       3,164
0       0       0       2,341
0       0       0       3,150
0       0       0       2,789
0       0       0       3,234
1       0       1       6,313
-1      0       -1      4,354
0       1       1       3,671
0       -1      -1      0,125

Stored 'archivo' (str)
```

Es evidente que podemos observar los datos experimentales registrados en la base de datos, pero no podemos disponer de manera directa de esa información. Para ello, debemos convertirlo en una variable interpretable por el sistema.

In [2]:

```
from App.Interprete import Data
Datos = Data(archivo)()
%store Datos
```

```
Stored 'Datos' (dict)
```

Por ejemplo...

Planteamiento del modelo

El planteamiento del modelo matemático requiere de un proceso de análisis de las variables de estudio (¿qué variables presentan una mayor influencia?). Para ello, empezamos por importar los datos del problema experimental, luego realizamos un estudio estadístico mediante una gráfica de distribución normal y diagrama de pareto; y, finalmente, planteamos *TODOS* los posibles modelos y escogemos el que mejor se adapta al fenómeno de estudio.

```
In [1]:
```

```
%store -r Datos
```

Estabilidad

En primer lugar, es importante determinar la estabilidad del experimento.

```
In [2]:
```

```
from App.Pretratamiento.Estabilidad import Est
Estabilidad = Est(Datos)

Estab = Estabilidad()
%store Estab
Estab
```

```
ADVERTENCIA: El coeficiente de variación (CV) está por enc
ima del 10%
Stored 'Estab' (dict)
```

```
Out[2]:
```

```
{'Promedio': 2.9356,
 'Desvest': 0.33532587135501485,
 'CV [%]': 11.422737135679755,
 'Varianza': 0.14055429999999994}
```

En caso de que existan datos por fuera, o por dentro, de la superficie del cubo experimental, se realiza la partición de datos para "Planteamiento" y "Validación".

Los datos para el **plantemiento** de los modelos matemáticos corresponden a los que se encuentran en la superficie del cubo (filas compuestas *únicamente* por: -1, 0 o 1); mientras que los datos de **validación** corresponden a los demás.

De maner resumida:

$$Plant = \begin{cases} 1 & \text{Nivel superior} \\ 0 & \text{Centro del cubo} \\ -1 & \text{Nivel inferior} \end{cases}$$

$$Val = \forall x (x \in \mathrm{R} \Lambda x \neq -1, x \neq 0, x \neq 1) x \in fila_{datos}$$

Para este caso particular...

```
In [3]:
```

```
Data_Plant = Estabilidad.Plant
Data_Val = Estabilidad.Val

%store Data_Plant
%store Data_Val

Data_Plant = Datos
```

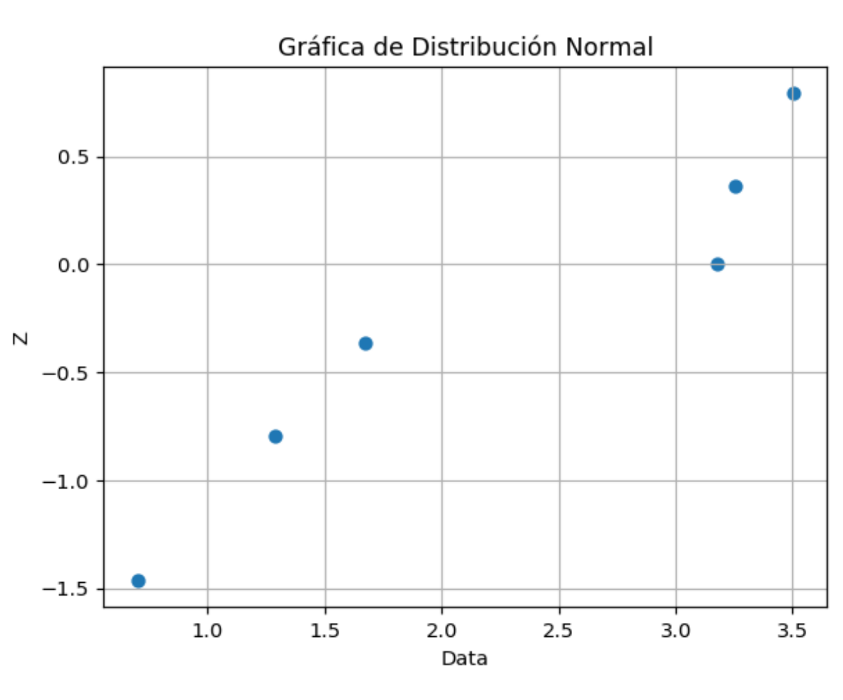
```
Stored 'Data_Plant' (dict)
Stored 'Data_Val' (dict)
```

Análisis de la varianza

A continuación, se puede apreciar la variación entre los datos experimentales a partir de una gráfica de distribución normal y un diagrama de pareto.

```
In [4]:
```

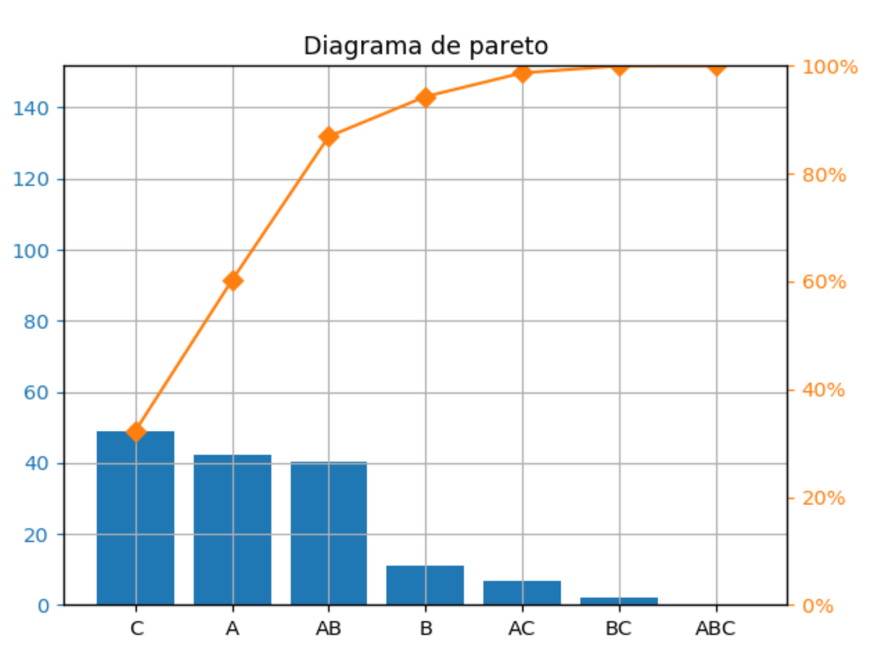
```
%matplotlib notebook
from App.Pretratamiento.ANOVA import NormalGraph, Pareto
NormDist = NormalGraph(Data_Plant)()
%store NormDist
```



```
Stored 'NormDist' (dict)
```

```
In [5]:
```

```
%matplotlib notebook
#Verdadero = True     Falso = False
P = Pareto(Data_Plant, porcentaje = 95, grid=True)
Efecto = P.ef
print('Variables de mayor efecto:', Efecto)
```



```
Variables de mayor efecto: ('C', 'A', 'AB', 'B')
```

Modelos Matemáticos

El análisis de la varianza nos ayuda a entender cuáles son las variables con mayor efecto dentro del fenómeno experimental estudiado. En pocas palabras, permite identificar un punto de partida para el inicio del proceso iterativo. Gracias a la tecnología, es posible evaluar cientos (y puede que miles) de modelos matemáticos en cuestión de segundos!

Pero antes de pensar en plantear múltiples modelos de manera simultánea, concentrémonos en plantear el primero.

Planteamiento

El modelo matemático general planteado es del orden:

$$Y = bX$$

Donde Y corresponde a la respuesta, b a las constantes y X a las variables.

Durante la práctica experimental, obtuvimos diversas variables de respuesta $\left(Y_{exp}\right)$, conocemos cuáles son las variables X, pero desconocemos el valor de las constantes, además del comportamiento propio del modelo que permite apreciar la solución "exacta" (experimental). Para calcular este valor, se desarrolla un modelo matemático que puede ser del tipo mostrado en la Ecuación 2.

$$Y_{cal} = B_0 + B_1X_1 + B_2X_1X_2 + B_3X_3^2 + \ldots$$

En términos matriciales, la Ecuación 2 es:

$$b = \left(X^TX\right)^{-1}Y_{exp}$$

Error inherente

El modelo planteado acarrea un error inherente, el cual es estimado a partir del cálculo de los coeficientes de determinación $\left(R^2 y R^2_{ajus}\right)$. Estos coeficientes se calculan a partir de las Ecuaciones 4 y 5.

$$R^2 = \frac{\sum_{i=1}^{n}\left(Y_i - \bar{Y}_{exp}\right)^2 - \sum_{i=1}^{n}\left(Y_i - Y_{cal_i}\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}_{exp}\right)^2}$$

$$R^2_{ajus} = \frac{\sum_{i=1}^{n}\left(Y_i - \bar{Y}_{exp}\right)^2/GL_{exp} - \sum_{i=1}^{n}\left(Y_i - Y_{cal_i}\right)^2/GL_{cal}}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}_{exp}\right)^2/GL_{exp}}$$

Dónde: Y_i corresponde a la respuesta experimental de la línea i del total del número de experimentos n, $\bar{Y}_{exp}$ corresponde a la media de las respuestas experimentales, Y_{cal_i} es la respuesta calculada por el modelo propuesto para la línea i, GL_{exp} corresponde a los grados de libertad experimentales y GL_{cal} a los grados de libertad del modelo.

Los grados de libertad se calculan a partir del número de datos n junto con el número de variables m, como se muestra a continuación:

$$GL_{exp} = n - 1$$

$$GL_{cal} = n - m - 1$$

Modelos Iniciales

Los modelos iniciales evaluados corresponden a las combinaciones posibles de las variables seleccionadas en la sección *"Análisis de la varianza"*.

```
In [6]:
```

```
from App.Modelos.Planteamiento import ModelosIniciales, ModeloFinal
from App.Modelos.Respuesta import *
nombre_bd = 'base'
P = ModelosIniciales(Data_Plant, NormDist, Efecto, nombre_bd)
Iniciales = P.Mejores
Mejor = P.Mejor
%store nombre_bd
```

```
Stored 'nombre_bd' (str)
```

```
In [7]:
```

```
for key in Iniciales:
    print('Modelo ' + str(key) + ':')
    Models(Iniciales[key])
```

```
Modelo 1:
```

$Y = \beta_0 + \beta_1C \rightarrow \left[\beta_0, \beta_1\right] = [4.574, 1.143]$

$R^2_{ajus} = 0.074$

```
Modelo 2:
```

$Y = \beta_0 + \beta_1A \rightarrow \left[\beta_0, \beta_1\right] = [4.574, 1.063]$

$R^2_{ajus} = 0.059$

```
Modelo 3:
```

$Y = \beta_0 + \beta_1AB \rightarrow \left[\beta_0, \beta_1\right] = [4.467, 1.65]$

$R^2_{ajus} = 0.057$

```
Modelo 4:
```

$Y = \beta_0 + \beta_1B \rightarrow \left[\beta_0, \beta_1\right] = [4.574, 0.547]$

$R^2_{ajus} = -0.01$

```
Modelo 5:
```

$Y = \beta_0 + \beta_1C + \beta_2A \rightarrow \left[\beta_0, \beta_1, \beta_2\right] = [4.574, 1.063, 0.976]$

$R^2_{ajus} = 0.122$

```
Modelo 6:
```

$Y = \beta_0 + \beta_1C + \beta_2AB \rightarrow \left[\beta_0, \beta_1, \beta_2\right] = [4.467, 1.143, 1.65]$

$R^2_{ajus} = 0.135$

```
Modelo 7:
```

$Y = \beta_0 + \beta_1C + \beta_2B \rightarrow \left[\beta_0, \beta_1, \beta_2\right] = [4.574, 1.106, 0.457]$

$R^2_{ajus} = 0.059$

```
Modelo 8:
```

$Y = \beta_0 + \beta_1A + \beta_2AB \rightarrow \left[\beta_0, \beta_1, \beta_2\right] = [4.467, 1.063, 1.65]$

$R^2_{ajus} = 0.12$

```
Modelo 9:
```

$Y = \beta_0 + \beta_1A + \beta_2B \rightarrow \left[\beta_0, \beta_1, \beta_2\right] = [4.574, 1.025, 0.463]$

$R^2_{ajus} = 0.044$

```
Modelo 10:
```

$Y = \beta_0 + \beta_1AB + \beta_2B \rightarrow \left[\beta_0, \beta_1, \beta_2\right] = [4.467, 1.65, 0.547]$

$R^2_{ajus} = 0.049$

```
Modelo 11:
```

$Y = \beta_0 + \beta_1C + \beta_2A + \beta_3AB \rightarrow \left[\beta_0, \beta_1, \beta_2, \beta_3\right] = [4.467, 1.063, 0.976, 1.65]$

$R^2_{ajus} = 0.188$

```
Modelo 12:
```

$Y = \beta_0 + \beta_1C + \beta_2A + \beta_3B \rightarrow \left[\beta_0, \beta_1, \beta_2, \beta_3\right] = [4.574, 1.034, 0.947, 0.385]$

$R^2_{ajus} = 0.103$

```
Modelo 13:
```

$Y = \beta_0 + \beta_1C + \beta_2AB + \beta_3B \rightarrow \left[\beta_0, \beta_1, \beta_2, \beta_3\right] = [4.467, 1.106, 1.65, 0.457]$

$R^2_{ajus} = 0.122$

```
Modelo 14:
```

$Y = \beta_0 + \beta_1A + \beta_2AB + \beta_3B \rightarrow \left[\beta_0, \beta_1, \beta_2, \beta_3\right] = [4.467, 1.025, 1.65, 0.463]$

$R^2_{ajus} = 0.106$

```
Modelo 15:
```

$Y = \beta_0 + \beta_1C + \beta_2A + \beta_3AB + \beta_4B \rightarrow \left[\beta_0, \beta_1, \beta_2, \beta_3, \beta_4\right] = [4.467, 1.034, 0.947,$

$R^2_{ajus} = 0.17$

Mejor modelo Inicial

El modelo base seleccionado es:

```
In [8]:
```

```
Models(Mejor)
```

$Y = \beta_0 + \beta_1C + \beta_2A + \beta_3AB \rightarrow \left[\beta_0, \beta_1, \beta_2, \beta_3\right] = [4.467, 1.063, 0.976, 1.65]$

$R^2_{ajus} = 0.188$

```
Out[8]:
```

```
<App.Modelos.Respuesta.Models at 0x2970b31f630>
```

Modelo Final

El modelo base corresponde al mejor modelo seleccionado con exponente a la 1. Ahora, se evaluará la misma combinación con diferentes exponentes. El criterio de selección es el mayor R^2_{ajus} posible. Se trata de un proceso iterativo en el que se evalúa la tendencia del criterio, para prever la mejor combinación de exponentes que permita seleccionar el modelo matemático final.

```
In [9]:
```

```
#Ecuación a evaluar
Porcentaje = 1.1        #Recomendable: 0.85 - Visualizar < 1.0
Eq = Mejor['Ecuación']
Eq = ('C', 'A', 'AB', 'ABC')
Final = ModeloFinal(Eq, NormDist, ref = 0.95, Y = Data_Plant['Y'],\
                    maximo=2, db='db', Porcentaje=Porcentaje)
```

```
In [ ]:
```

```
#¿Quieres seleccionar el mejor modelo inicial?
Modelo = Mejor
%store Modelo
```

```
In [10]:
```

```
Modelo = Final.Ans
Models(Modelo)
%store Modelo
```

$Y = \beta_0 + \beta_1C + \beta_2A + \beta_3A^2 + \beta_4AB^2 + \beta_5ABC^2 + \beta_6AB^2C^2 \rightarrow \left[\beta_0, \beta_1, \beta_2, \beta_3,\right.$

$R^2_{ajus} = 0.959$

```
Stored 'Modelo' (dict)
```

Resultados Gráficos

Los resultados gráficos (Y vs Ycal y Residuo) se pueden apreciar a continuación.

```
In [11]:
```

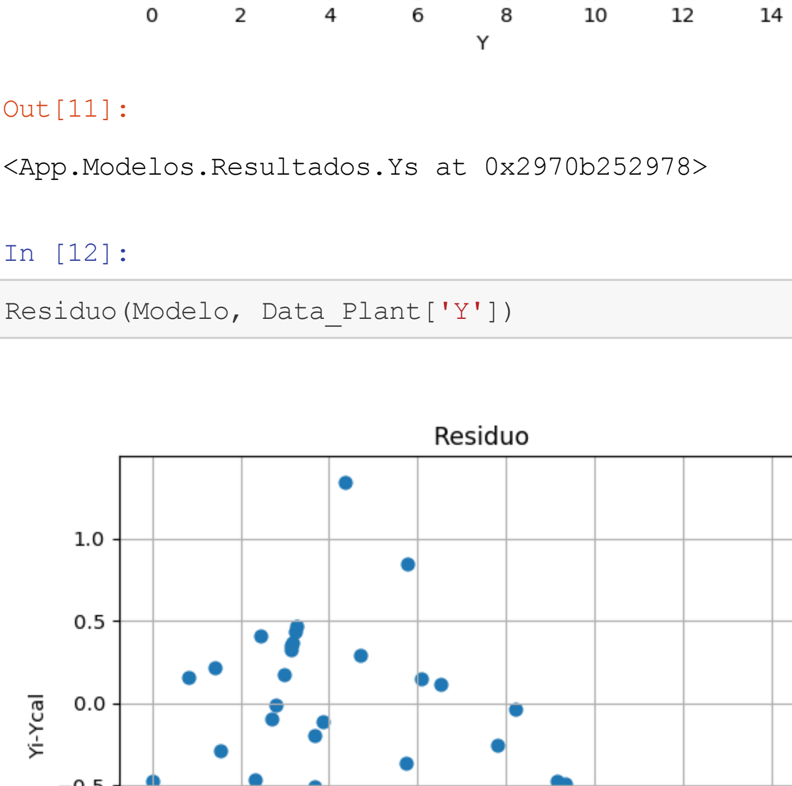
```
from App.Modelos.Resultados import *
Ys(Modelo, Data_Plant['Y'])
```

```
Out[11]:
```

```
<App.Modelos.Resultados.Ys at 0x2970b252978>
```

```
In [12]:
```

```
Residuo(Modelo, Data_Plant['Y'])
```



```
Out[12]:
```

```
<App.Modelos.Resultados.Residuo at 0x2970b9d49b0>
```

```
In [13]:
```

```
#Para ejemplo 1
limits = {
    'A':{
        '-1':3,
        '1':8
    },
    'B':{
        '-1':30,
        '1':60
    },
    'C':{
        '-1':8,
        '1':20
    }
}

SuperficieRespuesta(Modelo, limits)
```

VALIDACIÓN DEL MODELO

La validación del modelo matemático es una etapa crucial dentro del diseño experimental debido a que garantiza tanto la viabilidad como el dominio de operación del modelo que describe el fenómeno físico-químico de estudio.

El planteamiento de la Relación Cuantitativa Estructura - Propiedad (QSPR, por sus siglas en inglés) conlleva las siguientes prácticas:

Prácticas comunes:

1. **Preparación de los datos experimentales:** Incluye (i) la recolección y *limpieza* de los datos; (ii) cálculo de los decriptores químicos que contengan las propiedades objetivo; y (iii) unión de la propiedad y la respuesta en una base de datos de tipo SPR.
2. **Generación del modelo:** Implica establecer relaciones estadísticas entre las propiedades objetivo y la respuesta experimental.

   *Prácticas **no** comunes:*

3. **Validación del modelo:** Implica la evaluación cuantitativa de la solidez del modelo y su capacidad predictiva.
4. **Definición del dominio de aplicación:** Dominio matemático donde es viable emplear el modelo propuesto.

In [1]:

```
%store -r Modelo
%store -r Data_Val
%store -r Data_Plant
%store -r nombre_bd
%store -r Estab
%store -r Datos
%store -r archivo
```

Metodologías de validación

Existen diversas metodologías de validación, entre ellas:

- **LMO:** La premisa de esta metodología está en que si un modelo QSPR tiene una media alta q^2 durante el proceso de validación, se puede concluir que el modelo obtenido es robusto. Los n datos se dividen en G grupos de igual tamaño, $m_j(= n/G)$. Dependiendo del valor de n, G generalmente se selecciona entre 2 y 10. Un gran grupo de modelos son desarrollados con cada $n - m_j$ objetos en el conjunto de entrenamiento y m_j objetos en el conjunto de validación. Para cada modelo correspondiente, se predicen los objetos m_j y se computa q^2. Son deseables altos valores de q^2.
- **Bootstrapping:** Metodología de remuestreo que funciona con una muestra representativa de la población. Como sólo hay un grupo de datos, el bootstrapping simula qué ocurriría si las muestras se seleccionaran de manera aleatoria. En un procedimiento de validación típico, se generan k grupos aleatorios de tamaño n del grupo de datos original. Al igual que la validación LMO, es deseable un alto q^2 que demuestre la robustez del modelo.
- **Prueba de aleatoriedad - Y:** Técnica ampliamente usada para garantizar la robustez del modelo QSPR. En esta prueba, el vector de variables dependiente, Y, se cambia de manera aleatoria para la generación de un nuevo modelo QSPR usando la matriz original de variables independientes. El proceso se repite en varias ocasiones. Se espera que los modelos QSPR resultantes tengan bajos valores de R^2 y LOO (Leave-One-Out) q^2. Es probable, aunque infrecuente, que se obtengan altos valores q^2 debido a una correlación fortuita o una redundancia estructural del conjunto de datos. Si todos los modelos QSPR obtenidos por aleatoriedad - Y presentan altos valores de R^2 y q^2, implica que un modelo QSPR aceptable **no** puede ser obtenido por el grupo de datos dado.
- **Validación externa:**
  - *Selección de datos de planteamiento y validación:* En situaciones típicas, es difícil encontrar nuevos compuestos probados experimentalmente para este propósito. El recurso empleado es, entonces, dividir el conjunto de datos experimentales en datos para el plantemiento del modelo QSPR y datos para validación externa. El objetivo en esta etapa es garantizar que tanto los datos de validación como los de planteamiento ocupen el mismo dominio del fenómeno físico-químico de estudio. La partición de datos entre planteamiento y validación es un área de investigación activa. Es recomendable que el conjunto de validación externa contenga al menos 5 compuestos que describan el rango de actividad de los compuestos incluidos en el conjunto de planteamiento. Está demostrado que modelos QSPR desarrollados y validados mediante esta metodología tiene un alcance predictivo mayor que los mencionados anteriormente.
  - *Evaluación del poder predictivo de modelos QSPR:* En orden de estimar la verdadera capacidad predictiva del modelo QSPR, es necesario comparar las actividades predecibles y observadas de una base de datos lo suficientemente grande que no hubiese sido empleada para el desarrollo del modelo matemático. El poder predictivo del modelo QSPR se puede estimar mediante un q^2 externo definido mediante la Ecuación 1.

    $$q^2_{ext} = 1 - \frac{\sum_{i=1}^{Pr}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{Pr}\left(y_i - \bar{y}_{tr}\right)^2}$$

    Dónde y_i y $\hat{y}_i$ son los valores medido y predecido, respectivamente, de la variable dependiente $\bar{y}_{tr}$, que es el valor medio del conjunto de datos.

    Se considera que el modelo es **predictivo** cuando se satisfacen las siguientes condiciones: $$ \begin{equation} \begin{array}

    ```
    q q^2 > 0.5\\ \\
    R ^2 > 0.6 \\ \\
    \frac{\left(R^2 - R ^2 _0 \right)}{R^2} < 0.1 \\ \\
    0.85 \leq k \leq 1.15
    ```

    \end{array} \end{equation} $$

La metodología de validación empleada a continuación es la **prueba de aleatoriedad - Y**.

In [15]:

```
from App.Validation.AlY import AleatoriedadY
from App.Validation.Results import TablaAlY
AlY = AleatoriedadY(Modelo, Datos['Y'], Estab, nombre_bd)
ValData = AlY.Respuesta
TablaAlY(ValData, Modelo['Ecuación'])
```

R^2_{cal}	R^2_{val}	$RMSEC$	$RMSEP$	LOF	β_0	β_1	β_2	β_3	β_4	
0.981	0.932	0.542	2.936	9.337	2.794	1.083	1.075	2.344	4.047	2.
0.983	0.685	0.544	12.208	41.483	2.786	1.101	1.045	2.306	4.031	2.
0.968	0.997	0.802	0.198	4.054	2.742	1.039	1.046	2.292	3.972	2.

Out[15]:

```
<App.Validation.Results.TablaAlY at 0x210a55b2080>
```

Definición del dominio de aplicación

No importa cuán robusto sea el modelo QSPR validado: *no es confiable pensar que los resultados del modelo apliquen para todo el universo químico.*

In [ ]:

```
```

Guardar

Los resultados, tanto los del desarrollo del modelo matemático como los de validación, se guardarán y los podrás encontrar en la dirección: App/DataBase/Resultados

In [16]:

```
from App.DataBase.Guardar import Save
Save(Modelo, ValData, Datos['Y'], archivo)
```

Out[16]:

```
<App.DataBase.Guardar.Save at 0x210a55bcac8>
```