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:

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

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

Para este caso particular...

In [3]:

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

%store Data_Plant
%store Data_Val

Data_Plant = Datos
```

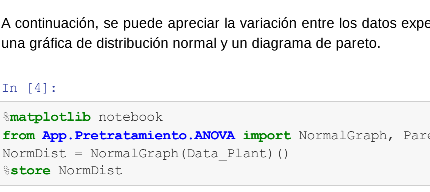
```
Stored 'Data_Plant' (dict)
Stored 'Data_Val' (dict)
```

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]:

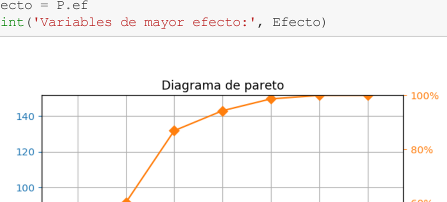
```
%matplotlib notebook
from App.Pretratamiento.ANOVA import NormalGraph, Pareto
NormDist = NormalGraph(Data_Plant)()
%store NormDist
```




```
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.

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

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

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
```

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

$R^2_{ajus} = 0.959$

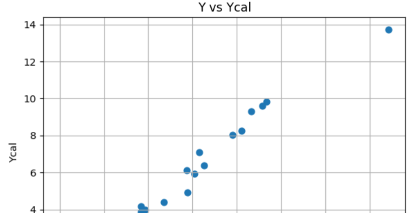
```
Stored 'Modelo' (dict)
```

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'])
```

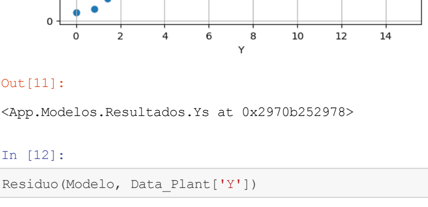


Out[11]:

```
<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)
```

Processing math: 100%