

Análisis de Datos

Métodos Numéricos y
Simulación.

Segundo de Grado en Física.

Motivación

Hemos hecho un experimento en el que hemos medido una magnitud Y y hemos encontrado que depende de una magnitud X

Ejemplo: hemos medido la diferencia de potencial V entre los extremos de una barra de cobre en función de la intensidad de corriente I que circula por la barra.

La corriente I está impuesta por nosotros, es decir, podemos elegir cuánto vale. Hace las veces de magnitud X y la llamamos variable **independiente**.

La diferencia de potencial V no la podemos controlar, simplemente la medimos. Hace las veces de magnitud Y y la llamamos variable **independiente**.

Tabla de ejemplo

Datos para la caída de tensión en una barra de cobre en función de la intensidad
 La primera fila es la intensidad en amperios
 Las columnas son la caída de potencial en micro-voltios
 Resultados de TEII y FISII, curso 2009-2010.

Intensidad (A)	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
d.d.p. (micro-v)										
2.1	4.2	6.6	9	11.2	13.5	15.8	17.9	19.8	22.3	
1	3	6	8	10	12	14	17	19	21	
3	5	7.3	9	10.9	13.7	16.2	18.2	20.7	22.8	
2.5	4.4	7	9	11.3	13.1	15.5	17.4	20	22.1	
0.4	2.82	5.43	7.3	9.72	13.37	15.81	18.2	21.7	24.1	
0.5	2.5	4.8	7.5	10	12.3	14.1	16.7	21.3	23.5	
0.5	3	5.3	7.4	9.8	12.3	14.4	16.7	19	21.2	
1.5	3.1	5.9	8.1	10.7	13.2	15	17.1	19.4	22.3	
1.7	3.9	6.3	8.4	10.7	13.1	15.2	17.4	19.8	21.9	
3.5	6	8.1	10.5	12.5	15.2	16.9	19.1	22.2	24.2	
3	4.9	6.9	8.6	11.3	13.8	16.1	18.6	21.1	24.1	
2	4.3	6.4	9.1	10.9	13.7	16	18	19.8	22.3	
2.4	4.9	7.3	9.7	11.9	13.9	16.1	18.3	20.6	23	
2.8	5	7.1	9.4	11.4	13.5	15.5	17.8	20.1	22.2	
2	4	6	9	11	13	15	18	20	22	
1.5	4	6	7.9	10.7	13.1	15.4	17.4	19.7	21.9	
3	5.3	7.5	9.3	11.8	13.8	16.4	18.7	20.6	22.4	
2.3	4.5	6.9	8.7	11	13.4	15.7	18	20.2	22.6	
4	6.5	8.7	10.8	12.8	15	17	20	21.8	24	
1.8	4.5	7	9	11	13	16	18	20	23	
2	4.4	6.5	8.8	11	12.8	15.3	17.5	19.5	21.8	
3.5	6.2	8.5	10.5	13	15.5	17.7	20.5	22.5	24.9	
3.1	5.2	7.3	9.7	11.7	13.7	16.1	18.2	20.6	22.7	
1.3	3.4	5.6	7.6	10.1	12.5	14.6	16.9	18.7	21.2	
3.5	4.6	6.8	9	10.2	12	14.5	17.5	19.6	22.7	
3	5.4	8	9.8	12.4	14.4	16.3	18.5	20.8	23	
2.8	5.2	7.2	9.3	11.4	14	16.2	18.2	20.3	22.8	
2.5	4.8	7.1	9.3	11.3	13.7	16	18.3	20.2	22.3	
2.3	5	7	9.6	11.6	13.6	16	18.2	20.4	22.1	
2.3	5.5	7.6	10.5	12.9	13.8	15	18.5	21	22.6	
1.9	3.4	6.5	8.2	10.8	14	16.3	18.5	20.9	22.9	
3	5.3	7.6	9.8	12.3	14.1	16.9	19.2	21.9	23.7	
3.1	6	8.4	10.9	13.3	15.5	17.7	20.2	22.5	24.7	
1.5	3.7	5.8	7.9	10.2	12.3	14.6	16.8	19	21.2	

Estos datos se importan ejecutando el script: ImportarTB

Ajuste lineal

Para nuestro experimento, tenemos que tener un modelo $y=f(x, a_1, a_2, \dots, a_N)$ que describe como depende la variable Y de la variable X

En nuestro caso, el modelo es: $V = IR$

luego: $f(x; a_1, a_2) = a_2 x + a_1$

Y hacemos las identificaciones $x \rightarrow I$ $y \rightarrow V$ $a_2 \rightarrow R$

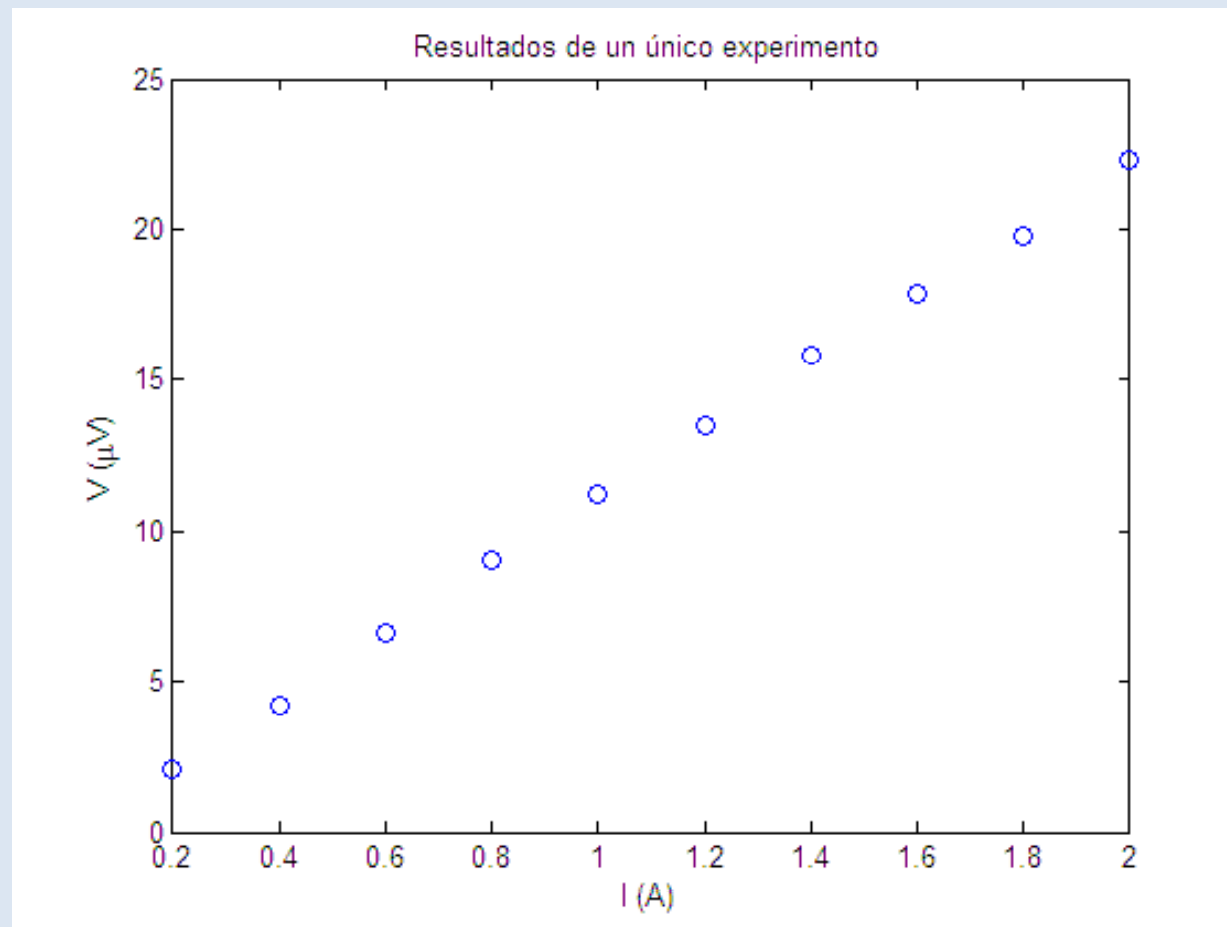
Esperamos que a_2 sea próximo al valor real R_{true} de la resistencia de la barra y que a_1 sea próximo a cero.

Ejemplo de ajuste lineal

Supongamos que representamos los datos del primer experimento.

```
V1=V(1,:);
```

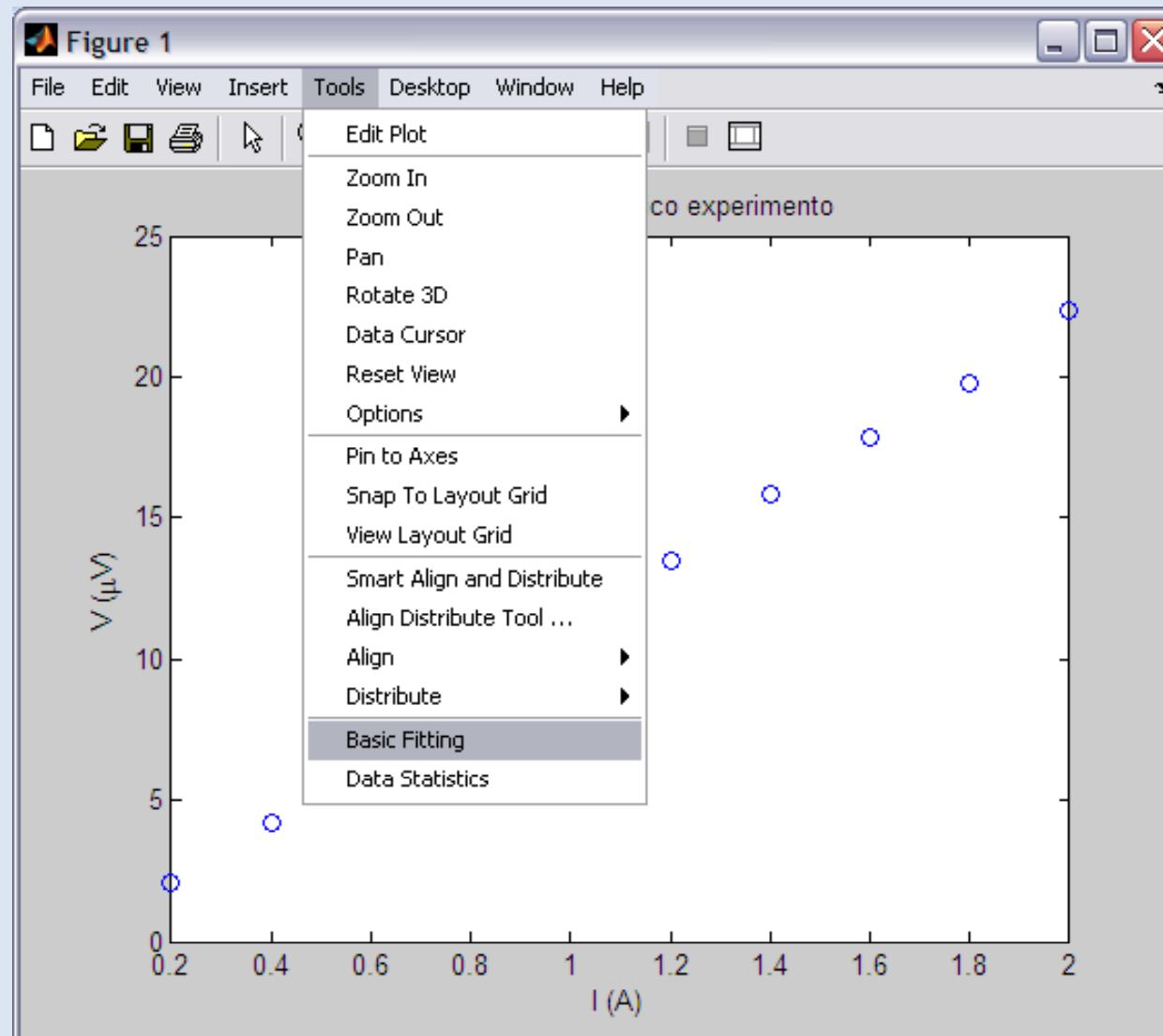
```
V1=V1';          plot(I,V1,'o');
```



Los resultados se aproximan bastante a una recta.

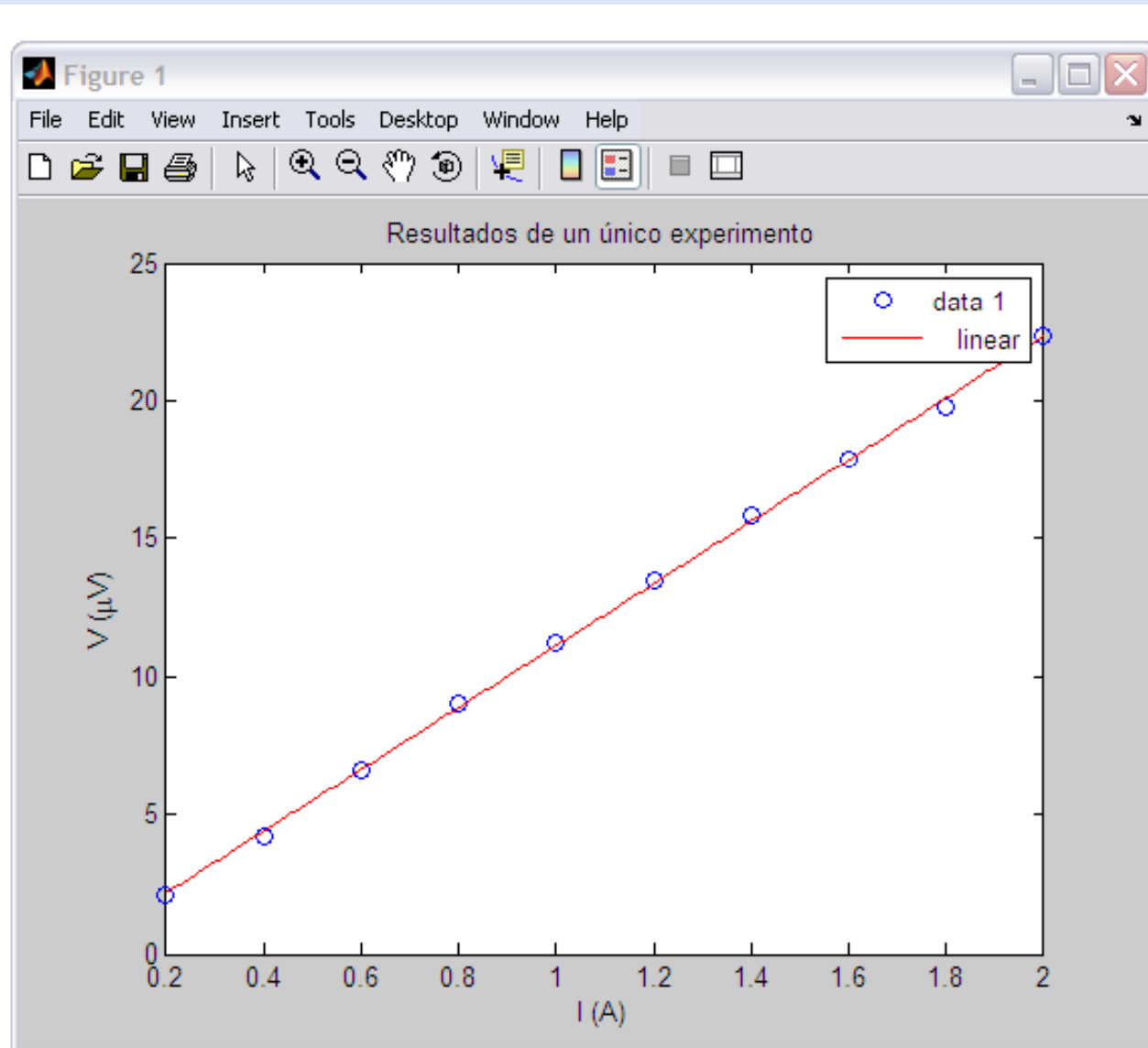
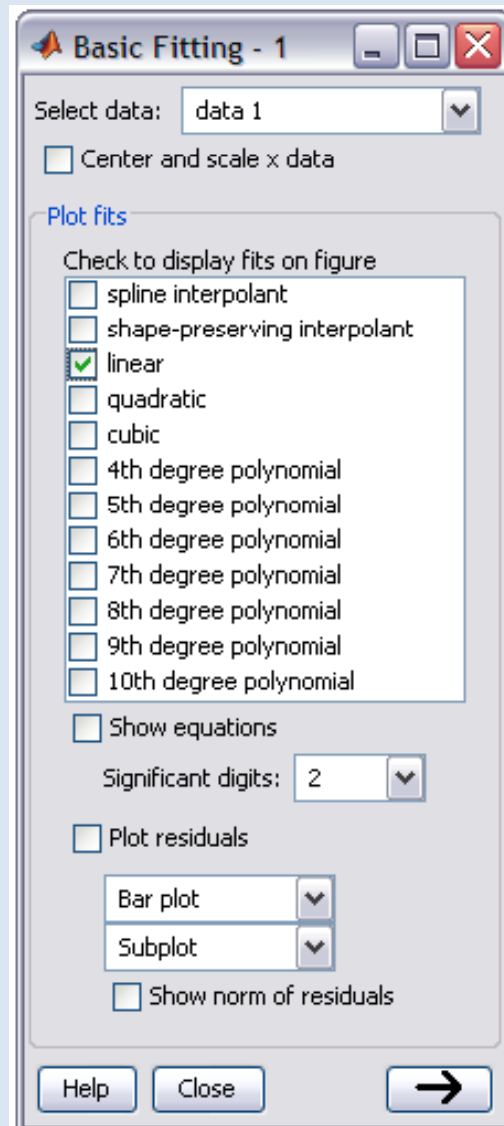
El menú “Basic Fitting”

Podemos ajustar una recta a los datos desde el menú *Tools/Basic Fitting*...



El menú "Basic fitting"

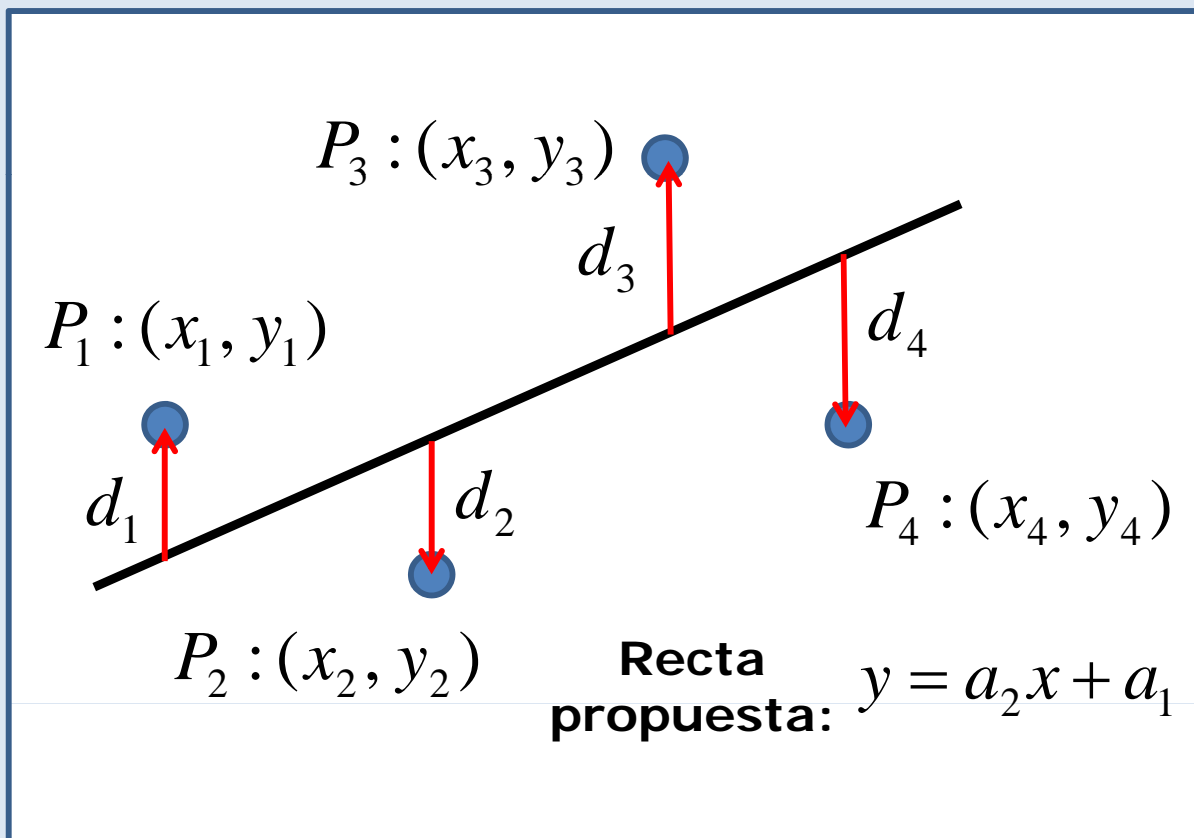
Y seleccionando la opción "linear"



Residuos

Hay que definir una función mérito que mida el acuerdo entre los datos experimentales y la función que queremos ajustar.

Como los puntos no están perfectamente alineados en una recta, si proponemos una recta cualquiera, habrá cierta distancia en vertical d_i entre cada punto i y la recta que hemos propuesto.



A las distancias d_i se las llama **residuos**.

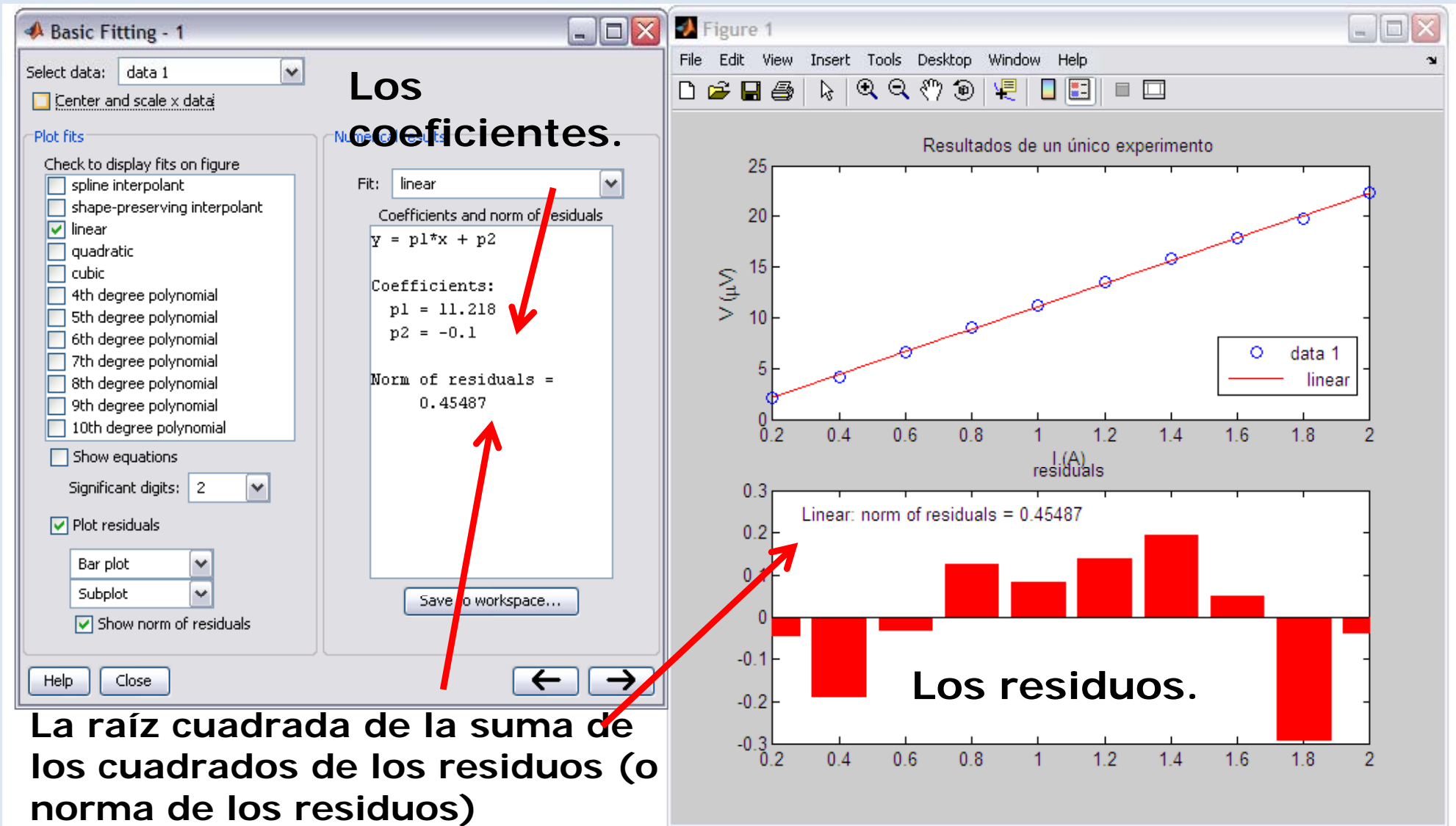
Por ejemplo, para d_3

$$d_3 = y_3 - (a_2x + a_1)$$

Los residuos pueden ser positivos o negativos.

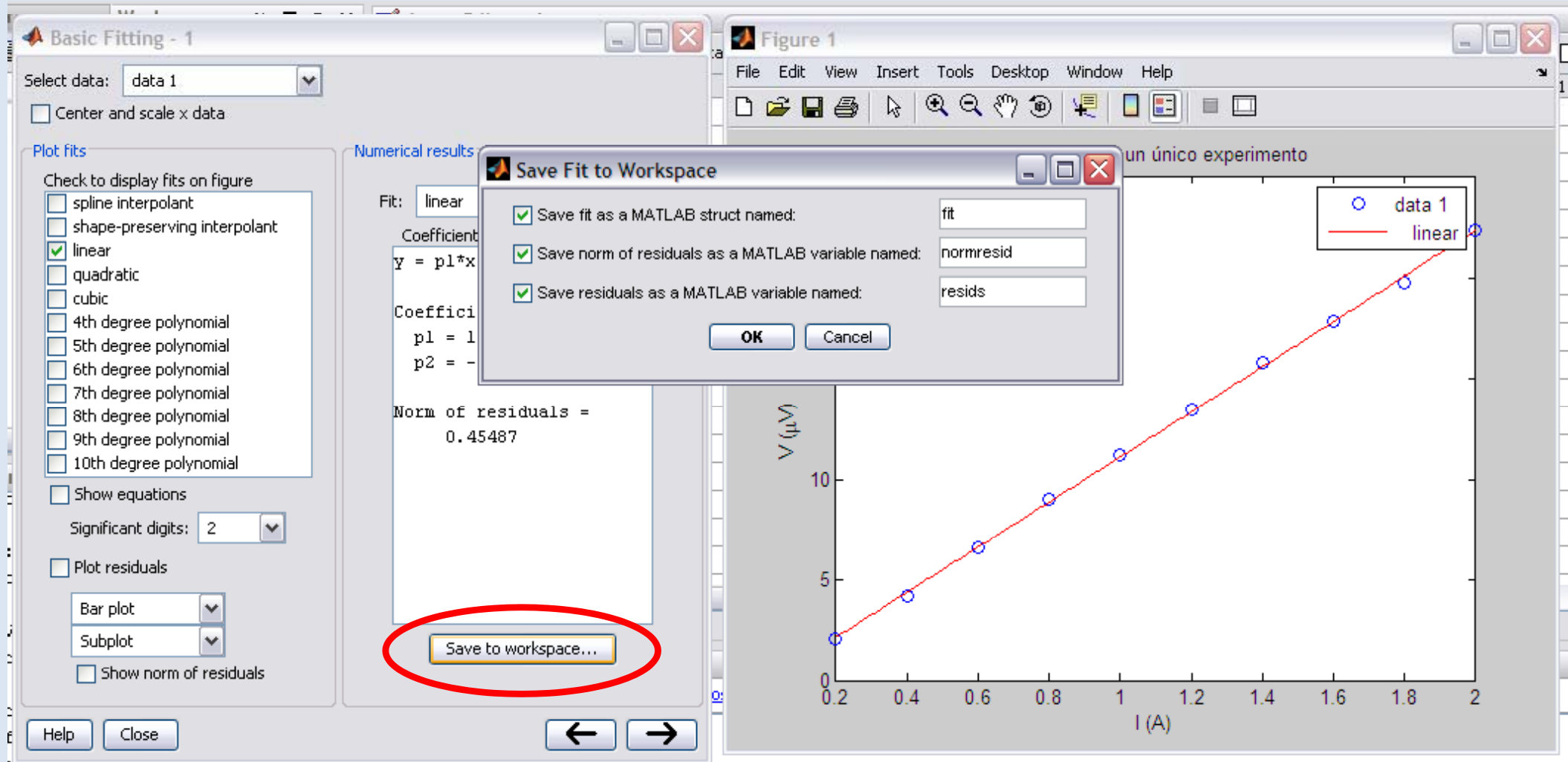
Visualización de los residuos

Una vez resueltas estas ecuaciones (que Matlab hace por nosotros), Matlab nos proporciona:

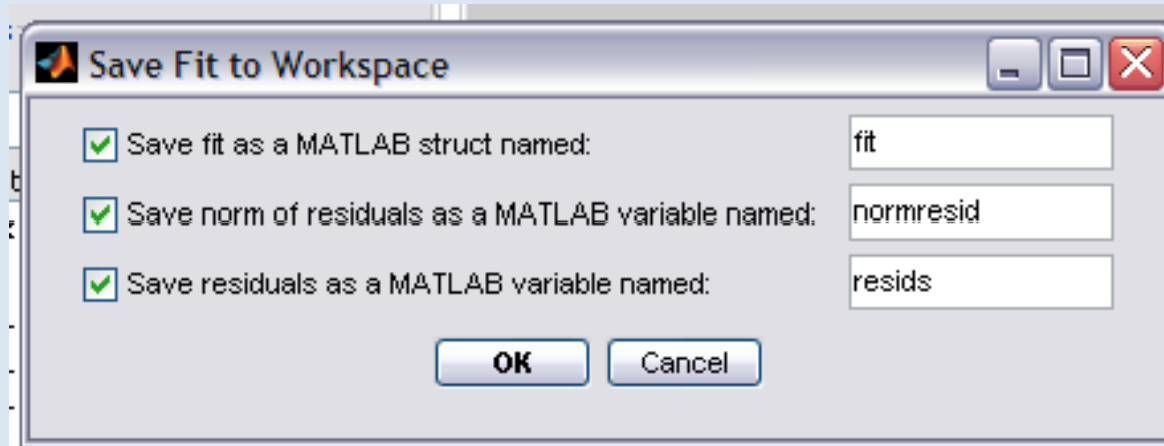


Exportar los resultados

Para grabar los resultados en el espacio de trabajo, se pica en el botón "Save to workspace" y se dan los nombres que queremos a las variables que se exportan.



¿Qué resultados se exportan?



- Una estructura que contiene el tipo de ajuste y los coeficientes.
- Una variable con la norma de los residuos
- Un vector con los valores de los residuos.

Estas variables aparecen en el espacio de trabajo (workspace) de Matlab cuando picamos en OK.

Ejemplo

The screenshot displays the MATLAB 7.4.0 (R2007a) interface. The workspace window shows a list of variables, with the 'fit' variable circled in red. The 'fit' variable is a 1x1 structure. The Array Editor window shows the fields of the 'fit' structure: 'type' is 'polynomial degree 1' and 'coeff' is '[11.218 -0.1]'. The Command History window shows the execution of the following commands:

```
[p S]=polyfit(I,V1',1);
clear
V1=V(1,:);
[p S]=polyfit(I,V1',1);
S.R
plot(I,V1);
r=CoficienteCorrelacion(I,V1);
V1=V1';
r=CoficienteCorrelacion(I,V1);
corrcoef(I,V1)
r=CoficienteCorrelacion(I,V1);
corrcoef(I,V1)
-- 1/10/10 17:22 --%
pepe.nombre = 'Pepe';
pepe.edad = 31;
-- 3/10/10 11:17 --%
V1=V(1,:);
```

The Command Window shows the execution of the command `V1=V(1,:);` and the message: "Variables have been created in the current workspace."

Aquí muestro los campos de la estructura fit. Dentro de unas transparencias veremos qué es una estructura.

La función *polyfit*

Si queremos hacer esta cuenta por línea de comando o en un programa, podemos usar la función ***polyfit***.

$$[p \ S] = \text{polyfit}(x,y,n)$$

Parámetros de entrada:

x: Vector con los valores de la variable independiente.

y: Vector con los valores de la variable dependiente.

n: orden del polinomio que se ajusta.

Parámetros de salida:

p: vector con los valores de los parámetros resultantes del ajuste.

S: Estructura que contiene otros resultados del ajuste.

Ejemplo: `teclead [p S] = polyfit(I,V1,1)`

Ejemplo

The image shows the MATLAB 7.4.0 (R2007a) interface. The main window is titled "MATLAB 7.4.0 (R2007a)" and contains several panes:

- Current Directory:** Shows the current directory as `F:\docencia\clases\materiales recurrentes\MNS\ analisis`.
- Workspace:** A table listing variables in the workspace:

Name	Value	Min	Max
I	[0.2;0.4;0.6;0.8;1;1...	0.2	2
S	<1x1 struct>		
V	<34x10 double>	0.4	24.9
V1	[2.1;4.2;6.6;9;11.2;...	2.1	22.3
fid	0	0	0
j	10	10	10
linea	'd.d.p. (micro-V)'		
p	[11.2182 -0.1]	-0.1	11.2
- Array Editor - S:** A table showing the fields of the structure `S`:

Field	Value
R	[-3.9243 -2.8031;0 -1.4639]
df	8
normr	0.45487
- Command History:** Shows the sequence of commands executed, including `av2=mean(q2);`, `av3=mean(q3);`, `av4=mean(q4);`, `av5=mean(q5);`, `[prob,t,sD]=tTest(q4,q5)`, `[prob,t,sD]=tTest(q1,q2)`, `clear`, `V1=V(1,:1)`, `V1=V(1,:)`, and `[p S]=polyfit(I,V1)`.
- Command Window:** Shows the output of the command `[p S]=polyfit(I,V1',1)`. The output is:

```
p =  
  
    11.2182    -0.1000  
  
S =  
  
      R: [2x2 double]  
      df: 8  
      normr: 0.4549
```

The text "Aquí muestro los campos de la estructura S que devuelve polyfit" is overlaid on the Array Editor pane.

Estructuras de datos

Una **estructura** es un tipo especial de contenedor de datos que alberga varios **campos**.

Cada campo puede contener datos de una naturaleza diferente.

Su símbolo en Matlab es:



A cada campo se accede escribiendo: Nombre-Estructura • Campo

Por ejemplo, en el caso de la estructura S que devuelve polyfit

S.R da la matriz R que resulta de la descomposición QR de la matriz de diseño del problema.

S.normr da la norma de los residuos

S.df da el número de puntos menos el número de parámetros del ajuste (df= número de grados de libertad)

El coeficiente de correlación

Tenemos que evaluar cómo de bueno es el ajuste. Para ello se define el coeficiente de correlación r (también llamado coeficiente de Pearson) como:

$$r = \frac{\sum_j (x_j - \langle x \rangle)(y_j - \langle y \rangle)}{\sqrt{\sum_j (x_j - \langle x \rangle)^2} \sqrt{\sum_j (y_j - \langle y \rangle)^2}}$$

Si $r = +/-1$, los datos se ajustan a una recta.

Si r es próximo a cero los datos no se ajustan a una recta.

Cálculo de r

En Matlab se calcula mediante la función ***Cr=corrcoef(x,y)***

Parámetros de entrada: un vector **x** con los valores de la variable independiente y un vector **y** con los valores de la variable dependiente.

Parámetros de salida: una matriz Cr con la forma:

$$Cr = \begin{pmatrix} 1 & r \\ r & 1 \end{pmatrix}$$

r es el coeficiente de correlación

Ejemplo: escribid `corrcoef(I,V1)`

Ajuste por mínimos cuadrados

La forma de encontrar la recta de mejor ajuste es encontrar los valores de a_2 y a_1 que minimizan la suma de los cuadrados de los residuos.

$$Norma^2 = \sum_{i=1}^N d_i^2 = \sum_{i=1}^N (y_i - a_2 x_i - a_1)^2$$

Para hallar el mínimo con lápiz y papel tenemos que hacer:

$$\frac{\partial Norma^2}{\partial a_2} = 0 \quad \Rightarrow \quad -2 \sum_{i=1}^N (y_i - a_2 x_i - a_1) x_i = 0$$

$$\frac{\partial Norma^2}{\partial a_1} = 0 \quad \Rightarrow \quad -2 \sum_{i=1}^N (y_i - a_2 x_i - a_1) = 0$$

Ecuaciones normales

Nos queda:

$$\sum_{i=1}^N (y_i x_i - a_2 x_i x_i - a_1 x_i) = 0 \quad \sum_{i=1}^N (y_i - a_2 x_i - a_1) = 0$$

A estas ecuaciones se las llama **ecuaciones normales** del ajuste. se suelen escribir de la forma.

$$\begin{aligned} S_{xy} &= a_2 S_{xx} + a_1 S_x & S_{xx} &= \sum_{i=1}^N x_i x_i & S_{xy} &= \sum_{i=1}^N x_i y_i \\ S_y &= a_2 S_x + a_1 S & S &= \sum_{i=1}^N 1 = N & S_x &= \sum_{i=1}^N x_i & S_y &= \sum_{i=1}^N y_i \end{aligned}$$

En forma matricial:

$$\begin{pmatrix} S_y \\ S_{xy} \end{pmatrix} = \begin{pmatrix} S & S_x \\ S_x & S_{xx} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad B = S \cdot A$$

A: Vector con los valores de los coeficientes

Solución de las ec. normales

Y su solución es:

$$\Delta = SS_{xx} - (S_x)^2 \quad a_2 = \frac{SS_{xy} - S_x S_y}{\Delta} \quad a_1 = \frac{S_{xx} S_y - S_x S_{xy}}{\Delta}$$

Y en forma matricial:

$$A = \mathbf{S}^{-1} \cdot B = \mathbf{C} \cdot B \quad \mathbf{C} = \mathbf{S}^{-1}$$

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} S_y \\ S_{xy} \end{pmatrix}$$

IMPORTANTE: Ni la matriz \mathbf{S} ni la matriz \mathbf{C} no dependen de y_1, y_2, \dots, y_N , porque en la matriz \mathbf{S} sólo aparecen sumas de potencias de x_1, x_2, \dots, x_N .

Solución seguida por Matlab

Volvamos al ajuste lineal. Las ecuaciones de la que lo obtuvimos son:

$$f(x; a_2, a_1) = a_2 x + a_1$$

$$0 = \sum_{i=1}^N (y_i - a_2 x_i - a_1) \quad 0 = \sum_{i=1}^N (y_i x_i - a_2 x_i x_i - a_1 x_i)$$

Podemos escribirlas como:

$$0 = \sum_{i=1}^N y_i \times 1 - \sum_{i=1}^N (a_2 x_i + a_1 \times 1) \quad 0 = \sum_{i=1}^N y_i x_i - \sum_{i=1}^N (a_2 x_i x_i - a_1 x_i)$$

Lo que he hecho ha sido simplemente separar por un lado las sumas que sólo contienen x's y las que contienen algún valor d las y's.

Ecuaciones en forma matricial

$$0 = \sum_{i=1}^N y_i \times 1 - \sum_{i=1}^N (a_2 x_i + a_1) \times 1$$

$$0 = \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix} - \begin{pmatrix} 1 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_N \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

$$0 = \sum_{i=1}^N y_i x_i - \sum_{i=1}^N (a_2 x_i - a_1) \times x_i$$

$$0 = \begin{pmatrix} x_1 & x_2 & \dots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix} - \begin{pmatrix} x_1 & x_2 & \dots & x_N \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_N \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

La matriz de diseño

Y ambas ecuaciones se pueden combinar como:

$$0 = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix} - \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_N \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

$$V^t(x_i) \cdot Y = V^t(x_i) \cdot V(x_i) \cdot A$$

V: matriz NxM llamada **matriz de diseño** del ajuste. M: número de parámetros. N: número de puntos.

Y: vector columna con los valores de la variable independiente

A: vector columna con los parámetros del ajuste.

¿Por qué calcular de esta forma?

Supongamos que ahora quiero ajustar N pares de datos experimentales por un polinomio de segundo orden:

$$f(x; a_1, a_2, a_3) = a_3 x^2 + a_2 x + a_1$$

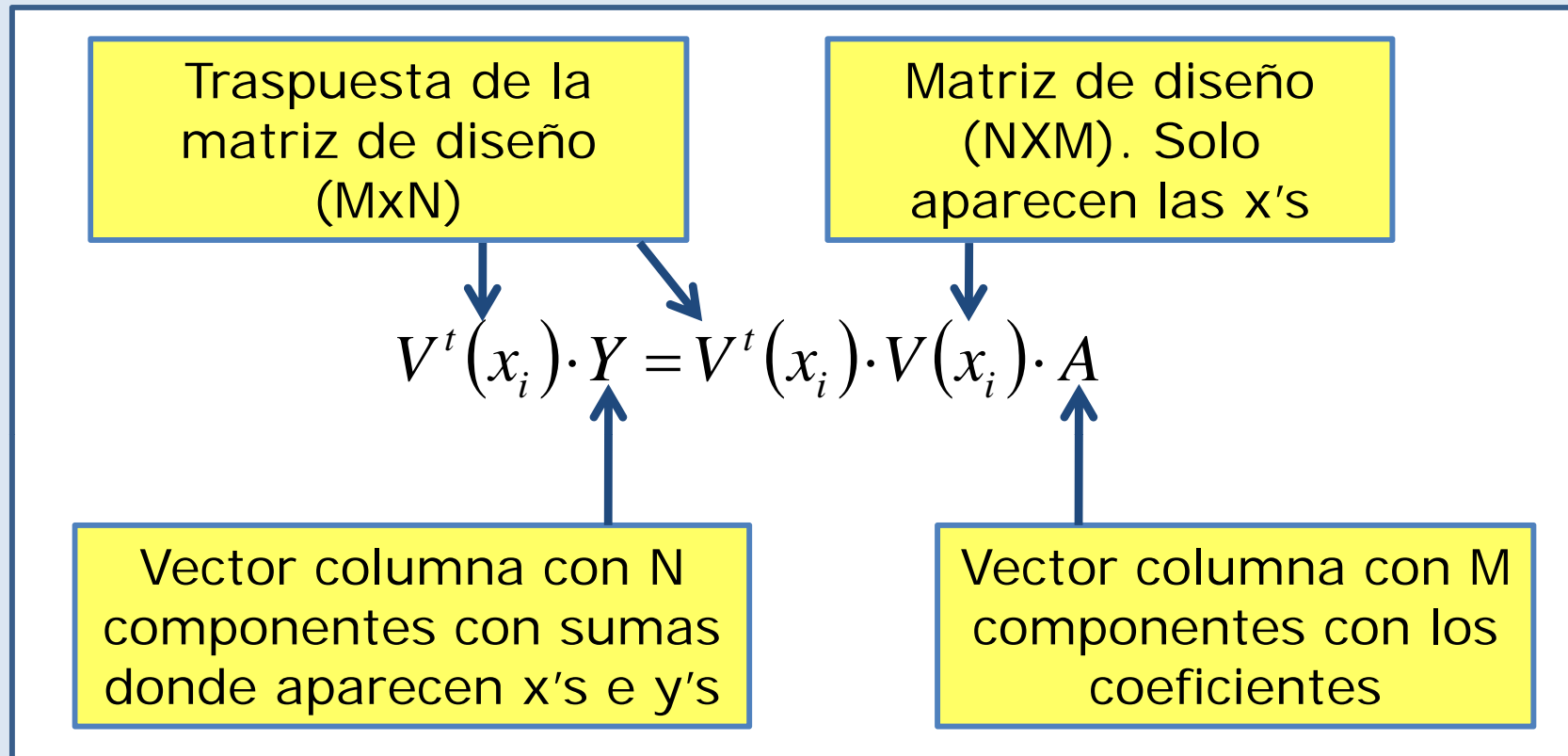
Repitiendo el procedimiento llego a un sistema de ecuaciones parecido

$$0 = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \\ x_1^2 & x_2^2 & \dots & x_N^2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix} - \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \\ x_1^2 & x_2^2 & \dots & x_N^2 \end{pmatrix} \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \dots & \dots & \dots \\ 1 & x_N & x_N^2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$

Puedo usar este procedimiento para polinomios de cualquier orden haciendo crecer la matriz de diseño hasta la potencia que desee.

Ajuste polinómico

En general, para ajustar por un polinomio de grado $M-1$:



N es el número de puntos

M es el número de parámetros.

Ajuste lineal generalizado

Supongamos que ahora quiero ajustar los datos por

$$f(x; a_1, a_2) = a_1 g(x) + a_2 h(x)$$

Donde $g(x)$, $h(x)$ y $w(x)$ son funciones conocidas que dependen exclusivamente de x .

Repitiendo el procedimiento llego al sistema de ecuaciones:

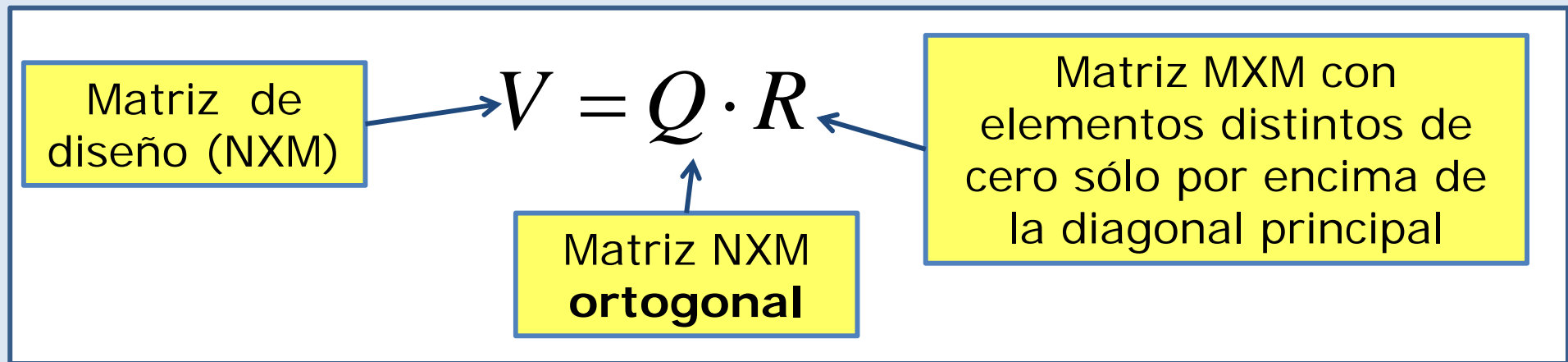
$$0 = \begin{pmatrix} g(x_1) & g(x_2) & \dots & g(x_N) \\ h(x_1) & h(x_2) & \dots & h(x_N) \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix} - \begin{pmatrix} g(x_1) & g(x_2) & \dots & g(x_N) \\ h(x_1) & h(x_2) & \dots & h(x_N) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

$$V^t(x_i) \cdot Y = V^t(x_i) \cdot V(x_i) \cdot A$$

Esta ecuación algebraica tiene la misma forma que hemos visto antes, aunque la matriz de diseño de calcula de forma diferente.

Descomposición Q-R

Siempre se puede descomponer la matriz de diseño en la forma:



Que la matriz Q sea ortogonal significa que:

$$Q^t \cdot Q = 1$$

La función de Matlab que hace esta descomposición es

$$[Q \ R] = \text{qr}(V)$$

Esta es la matriz R que aparece en la estructura S que devuelve `polyfit`.

Obtención de los parámetros

La ecuación que tenemos que resolver es

$$\begin{aligned} V^t \cdot Y &= V^t \cdot V \cdot A \\ \left(R^t \cdot Q^t \right) \cdot Y &= \left(R^t \cdot Q^t \right) \cdot (Q \cdot R) \cdot A = R^t \cdot R \cdot A \end{aligned}$$

Sacando factor común R^t :

$$R^t \cdot (Q^t \cdot Y - R \cdot A) = 0 \iff Q^t \cdot Y - R \cdot A = 0$$

Solución:

$$A = R^{-1} \cdot Q^t \cdot Y$$

A: Vector columna con M componentes que son los valores de los coeficientes

El algoritmo de *polyfit*

Este es el procedimiento que usa la función *polyfit(x,y,n)*

```
--  
52 - x = x(:);  
53 - y = y(:);  
54  
55 - if nargin > 2  
56 -     mu = [mean(x); std(x)];  
57 -     x = (x - mu(1))/mu(2);  
58 - end  
59  
60 % Construct Vandermonde matrix.  
61 - V(:,n+1) = ones(length(x),1,class(x));  
62 - for j = n:-1:1  
63 -     V(:,j) = x.*V(:,j+1);  
64 - end  
65  
66 % Solve least squares problem.  
67 - [Q,R] = qr(V,0);  
68 - ws = warning('off','all');  
69 - p = R \ (Q' * y); % Same as p = V \ y;
```

Escribe x,y como vectores columna

Construye la matriz de diseño

Resuelve las ecuaciones.

¿Eso es todo?

Un buen trabajo de ajuste debe proporcionar:

1) Los valores óptimos de los parámetros de ajuste.

En las transparencias anteriores hemos resuelto este punto.

2) Una medida de la bondad del ajuste, es decir, ¿describe nuestro modelo los datos experimentales?

Hemos resuelto este punto sólo para la regresión lineal, mediante el coeficiente de correlación.

3) La incertidumbre en los valores óptimos.

Este punto nos queda por resolver.

Incertidumbre en los parámetros

Retomamos el ajuste lineal:

$$f(x; a_2, a_1) = a_2 x + a_1$$

Sabemos que usando las ecuaciones normales a_1 y a_2 se encuentran de:

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} \begin{pmatrix} S_y \\ S_{xy} \end{pmatrix}$$

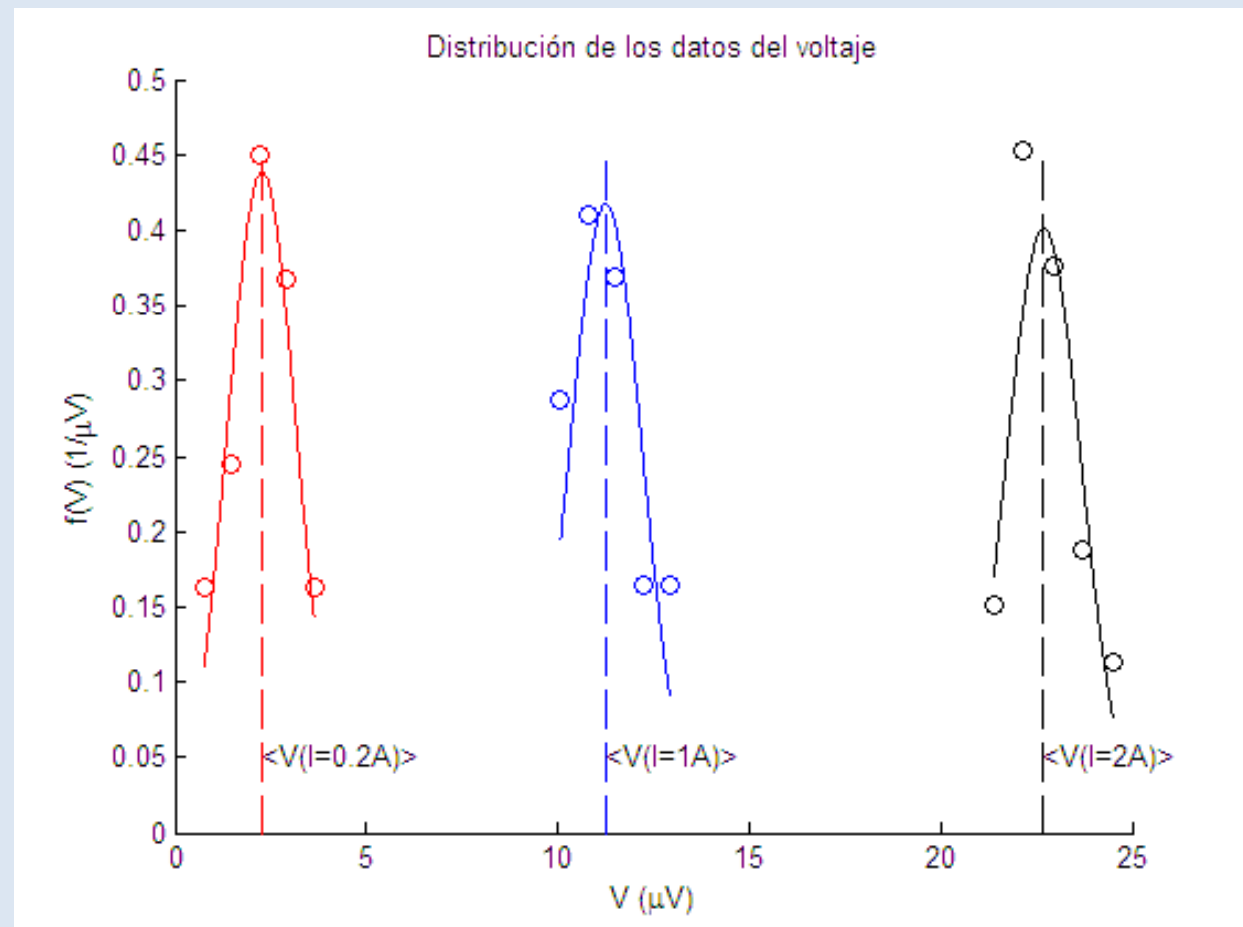
Suponemos que:

- x_1, x_2, \dots, x_N se conocen sin ningún tipo de incertidumbre, pero que manteniendo un valor de x , si se repite la medida de y se obtienen valores con una cierta distribución estadística.
- Para un x fijo, la distribución diferencial de los valores de y es una gaussiana con desviación estándar $\sigma(y)$
- La desviación estándar $\sigma(y)$ es independiente de los valores de x .

Ejemplo

Si tenemos en cuenta todos los experimentos en los que se ha medido la caída de potencial V en función de la intensidad I en la barra de cobre, resulta que para un voltaje fijo el potencial que se mide tiene una distribución estadística.

Como vemos en el ejemplo, los valores de la variable independiente del voltaje no tienen por qué seguir una distribución gaussiana. Sin embargo, normalmente se supone que es así.



Incertidumbre en a_1 (I)

Puesto que:

$$a_1 = C_{11}S_y + C_{12}S_{xy}$$

Si consideremos que a_1 es función de y_1, y_2, \dots, y_N , la fórmula de propagación de errores nos dice:

$$\sigma(a_1)^2 = \sum_{j=1}^N \sigma(y_j)^2 \left[\frac{\partial}{\partial y_j} (C_{11}S_y + C_{12}S_{xy}) \right]^2$$

$$\sigma(a_1)^2 = \sum_{j=1}^N \sigma(y_j)^2 \left[C_{11} \frac{\partial S_y}{\partial y_j} + C_{12} \frac{\partial S_{xy}}{\partial y_j} \right]^2$$

Puesto que C_{11} y C_{12} sólo dependen de x_1, x_2, \dots, x_N , y

$$\frac{\partial S_y}{\partial y_j} = \frac{\partial}{\partial y_j} \sum_{i=1}^N y_i = 1$$

$$\frac{\partial S_{xy}}{\partial y_j} = \frac{\partial}{\partial y_j} \sum_{i=1}^N x_i y_i = x_j$$

Incertidumbre en a_1 (II)

Sustituyendo llegamos a:

$$\sigma(a_1)^2 = \sum_{j=1}^N \sigma(y_i)^2 [C_{11} \times 1 + C_{12} x_j]^2$$

Y como todas las $\sigma(y_j)$ son iguales a $\sigma(y)$

Desarrollando el cuadrado.

$$\sigma(a_1)^2 = \sigma(y)^2 \sum_{j=1}^N [(C_{11})^2 \times 1 + (C_{12})^2 x_j^2 + 2C_{11}C_{12}x_j]$$

Sacando C_{11} y C_{12} factores comunes

$$\sigma(a_1)^2 = \sigma(y)^2 [(C_{11})^2 S + (C_{12})^2 S_{xx} + 2C_{11}C_{12}S_x]$$

Usando que $C = S^{-1}$

$$\sigma(a_1)^2 = \sigma(y)^2 [C_{11}(C_{11}S + C_{12}S_x) + C_{12}(C_{12}S_{xx} + C_{11}S_x)]$$

Resultado:

$$\sigma(a_1)^2 = \sigma(y)^2 C_{11}$$

Matriz de covariancias

En general, si hacemos un ajuste del tipo:

$$f(x; a_1, a_2, \dots, a_M) = a_M x^{M-1} + \dots + a_2 x + a_1$$

$$\sigma(a_j)^2 = \sigma(y)^2 C_{jj}$$

A la matriz $\mathbf{C} = \mathbf{S}^{-1}$ se la llama **matriz de covariancias**.

Pero `polyfit(x,y,n)` nos devuelve la matriz \mathbf{R} . ¿Qué hacemos para hallar \mathbf{C} ?

Cálculo de la matriz de covariancias

Comparando los dos procedimientos:

Ecuaciones
normales.

$$\begin{pmatrix} S_y \\ S_{xy} \end{pmatrix} = \begin{pmatrix} S & S_x \\ S_x & S_{xx} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

$$B = S \cdot A$$

$$0 = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{pmatrix} - \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{pmatrix} \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \dots & \dots \\ 1 & x_N \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

$$V^t(x_i) \cdot Y = V^t(x_i) \cdot V(x_i) \cdot A$$

Conclusión:

$$B = V^t \cdot Y \quad S = V^t \cdot V$$

Cálculo de la matriz de covariancias

Como sabemos que:

$$\mathbf{S} = \mathbf{V}^t \cdot \mathbf{V} \qquad \mathbf{V} = \mathbf{Q} \cdot \mathbf{R}$$

Sustituyendo:

$$\mathbf{S} = \mathbf{R}^t \cdot \mathbf{Q} \cdot \mathbf{Q}^t \cdot \mathbf{R} = \mathbf{R}^t \cdot \mathbf{R}$$

$$\mathbf{C} = \mathbf{S}^{-1} = \left(\mathbf{R}^t \cdot \mathbf{R}\right)^{-1} = \mathbf{R}^{-1} \cdot \left(\mathbf{R}^{-1}\right)^t$$

Y de esta forma encontramos la matriz de covarianzas. Por eso `polyfit(x,y,n)` sólo devuelve la matriz \mathbf{R} .

Recuerda que la inversa T de una matriz \mathbf{R} se calcula como $T = \mathit{inv}(\mathbf{R})$.

Estimación de la incertidumbre de los parámetros.

¿Qué hacer si no conocemos $\sigma(y)$?. Puede demostrarse que:

$$\sigma(a_j)^2 \approx C_{jj} \frac{Norma^2}{N - M}$$

N: número de puntos experimentales

M: número de parámetros

N-M: número de grados de libertad

$$Norma^2 = \sum_{i=1}^N d_i^2 \quad \text{cuadrado de la norma de los residuos}$$

Estimación de la incertidumbre de la variable dependiente.

Supongamos que hemos hecho un ajuste:

$$y = f(x; a_1, a_2, a_3) = a_3 x^2 + a_2 x + a_1$$

Y queremos evaluar el valor predicho para la variable y en un valor x_0 con su grado de incertidumbre. Para ello usamos la función:

$$[yo, delta] = polyval(p, xo, S)$$

Parámetros de entrada:

p: vector con los coeficientes del ajuste.

xo: valor de la variable independiente donde evaluamos el ajuste.

S: estructura devuelta por la función *polyfit*.

Parámetros de salida:

yo: valor predicho para la variable dependiente.

delta: intervalo de confianza para yo . El 50% de los valores experimentales para la variable independiente estará dentro de $(yo - delta, yo + delta)$.

Bondad del ajuste

Para el caso general, podemos dar la bondad del ajuste calculando:

$$\chi^2 = \frac{Norma^2}{\sigma(y)^2}$$

La probabilidad de haber obtenido **por azar** un valor de χ^2 igual o menor al que hemos obtenido es:

$$P = F_{chi-cuad}(\chi^2; \nu) = \text{gammainc}\left(\frac{\chi^2}{2}, \frac{\nu}{2}\right)$$

gammainc: función gamma incompleta.

N: número de puntos experimentales

M: número de parámetros de ajuste

$\nu = N - M$: número de grados de libertad.

Interpretación de la bondad del ajuste

P es la probabilidad de que, suponiendo que los valores de los parámetros a_1, a_2, \dots, a_M son correctos, los puntos (x_j, y_j) se ajusten **por azar** a la curva predicha por ellos teniendo en cuenta que los y_j tienen una distribución de anchura dada por $\sigma(y)$.

Por tanto, la probabilidad de **que el ajuste no se deba al azar** viene dada por:

$$Q = 1 - P = 1 - \text{gammainc}\left(\frac{\chi^2}{2}, \frac{\nu}{2}\right)$$

Resumen: Q bajo: mal ajuste. Q alto: buen ajuste