Diplomado : Herramientas de Programación para Ciencias e Ingeniería

Módulo: MATLAB (Clase 6)

Docente: Juan Sebastián Salcedo Gallo

Universidad Nacional de Colombia Sede Manizales

Contenido

- Funciones Polinomiales
- Interpolación
- Regresión

Funciones Polinomiales

Una función polinomial de grado n, generalmente se escribe como:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_2 x^2 + a_1 x + a_0$$

Las funciones polinomiales están definidas y son continuas en todos los números reales.

POLINOMIALES DE GRADO BAJO		
NOMBRE	**FORMA**	**GRADO**
Función constante	$f(x) = a$	0
Función lineal	$f(x) = ax + b,\ a \neq 0$	1
Función cuadrática	$f(x) = ax^2 + bx + c,\ a \neq 0$	2
Función cúbica	$f(x) = ax^3 + bx^2 + cx + d,\ a \neq 0$	3
Función cuártica	$f(x) = ax^4 + bx^3 + cx^2 + dx + e,\ a \neq 0$	4

Funciones Polinomiales MATLAB

Un polinomio se representa por un vector fila con sus coeficientes en orden descendente, no debemos olvidar los términos con coeficiente nulo.

Así por ejemplo si queremos indicar el polinomio 5x4 + 2x2 – x + 7 = 0 escribiríamos [5 0 2 -1 7].

Para encontrar las raíces de un polinomio p, usaremos la función **roots(p).**

Funciones Polinomiales MATLAB

En caso de conocerse las raíces de un polinomio es posible construir el polinomio asociado mediante la función poly(r)

NOTA: MATLAB TRABAJA CON LOS POLINOMIOS COMO VECTORES FILA, Y CON LAS RAÍCES COMO VECTOR COLUMNA.

Ejemplo

```
>> p = [1 -9 13 9 -14];        % representa al polinomio x^4-9x^3+13x^2-9x-14

>> roots (p)                   % calcula sus raíces
ans =
    7.0000
   -1.0000
    2.0000
    1.0000

>> poly (ans)                  % devuelve el polinomio generado por  esas cuatro raíces
ans =
    1.0000   -9.0000   13.0000    9.0000  -14.0000
```

Funciones útiles para efectual operaciones entre Polinomios

Función	**¿Qué es?**
conv(p,q)	Multiplica los dos polinomios p y q
deconv (c, q)	Divide el polinomio c entre q
polyder(p)	Calcula la derivada del polinomio p
polyder(p, q)	Calcula la derivada del producto p*q
polyval(p,A)	Evalúa el polinomio p en todos los elementos de A

Ejemplos

```
>> p = [1 2 7];
>> q = [1 3 6];                % polinomios

>> c = conv (p,q)              % producto de los polinomios p y q
c =
    1     5    19    33    42

>> deconv (c,q)                % cociente de dividir el polinomio c entre el polinomio q
ans =
    1     2     7

>> polyder (p)                 % derivada del polinomio p
ans =
    2     2

>> polyder (p,q)               % derivada del producto de los polinomios p y q
ans =
    4    15    38    33
```

Ejemplos

```
>> polyval (p, [0 1 5] )          % evalúa el polinomio en 0, 1 y 5, es decir, halla p(0), p(1) y p(5)
ans =
    7    10    42

>> polyval (p, [0 1 2; -1 -2 -3; 4 0 7] )          % igual pero toma los valores de una matriz
ans =
     7    10    15
     6     7    10
    31     7    70
```

Ejercicio

Usando la función empleada en clases pasadas para encontrar las raíces de un polinomio de orden 2 encuentre las raíces del polinomio p. Luego compruébelo con la función roots(p).

$$p = x^2 + 12x + 7 = 0$$

Ejercicios

Efectúe la siguiente operación y calcule la derivada del polinomio resultante. Hágalo a mano, y compruébelo usando las herramientas vistas anteriormente.

$$(8x^7 + 24x^4 + 32x^3)(12x^3 - 7x^2 + 5)$$

Encuentre las raíces de ambos polinomios. Después, con las raíces, encuentre los polinomios originales.

Interpolación

En el subcampo matemático del análisis numérico, se denomina interpolación a la obtención de nuevos puntos partiendo del conocimiento de un conjunto discreto de puntos.

En ingeniería y algunas ciencias es frecuente disponer de un cierto número de puntos obtenidos por muestreo o a partir de un experimento y pretender construir una función que los ajuste.

Interpolación lineal

“Unir” cada uno de los puntos de un conjunto de datos ya sea obtenido de forma experimental o empírica y ajustar líneas rectas entre sus puntos, para obtener una función que “reproduzca” el comportamiento cualitativo del conjunto de datos y lograr hacer predicciones.

Para esto, se usa el comando

Syntax

```
yi = interp1q(x,Y,xi)
```

El cual retorna un arreglo unidimensional con los datos de la interpolación

Interpolación lineal

En el comando anterior, **Syntax**

```
yi = interp1q(x,Y,xi)
```

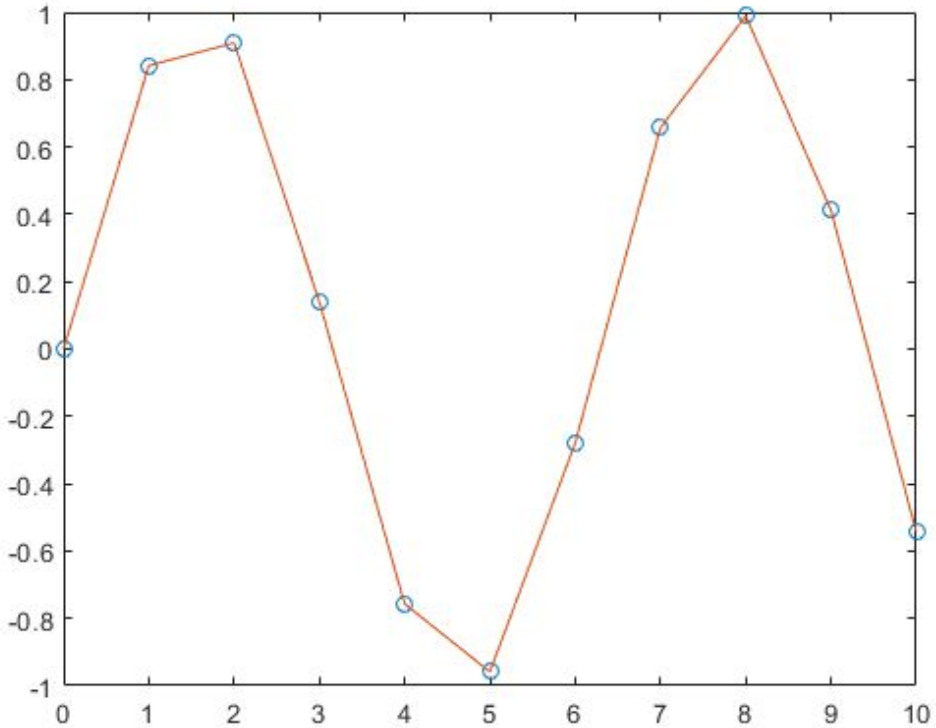
x debe ser un vector **columna** monotónicamente creciente.

Y debe ser un vector **columna** con un número de elementos length(x) filas.

xi debe ser un vector **columna**.

Ejemplo

```
x = (0:10)';
y = sin(x);
xi = (0:.25:10)';
yi = interp1q(x,y,xi);
plot(x,y,'o',xi,yi)
```



Interpolación por Spline Cúbico

El spline cúbico (k=3) es el spline más empleado, debido a que proporciona un excelente ajuste a los puntos tabulados y su cálculo no es excesivamente complejo.

Sobre cada intervalo [t0, t1], [t1,t2], .. ,[t(n-1),t(n)], S está definido por un polinomio cúbico diferente. Sea Si el polinomio cúbico que representa a S en el intervalo [ti, ti+1], se garantiza continuidad.

$$S(x) = \begin{cases} S_0(x) & x \in [t_0, t_1) \\ S_1(x) & x \in [t_1, t_2) \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1}, t_n) \end{cases}$$

Spline Cúbico

$$\begin{pmatrix} u_1 & h_1 & & & & \\ h_1 & u_2 & h_2 & & & \\ & h_2 & u_3 & h_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & h_{n-3} & u_{n-2} & h_{n-2} \\ & & & & h_{n-2} & u_{n-1} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_{n-2} \\ z_{n-3} \end{pmatrix} = \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \vdots \\ \nu_{n-2} \\ \nu_{n-1} \end{pmatrix}$$

Spline Cúbico en MATLAB

Para realizar interpolación por spline cúbico a un conjunto de datos, en MATLAB se utiliza el comando

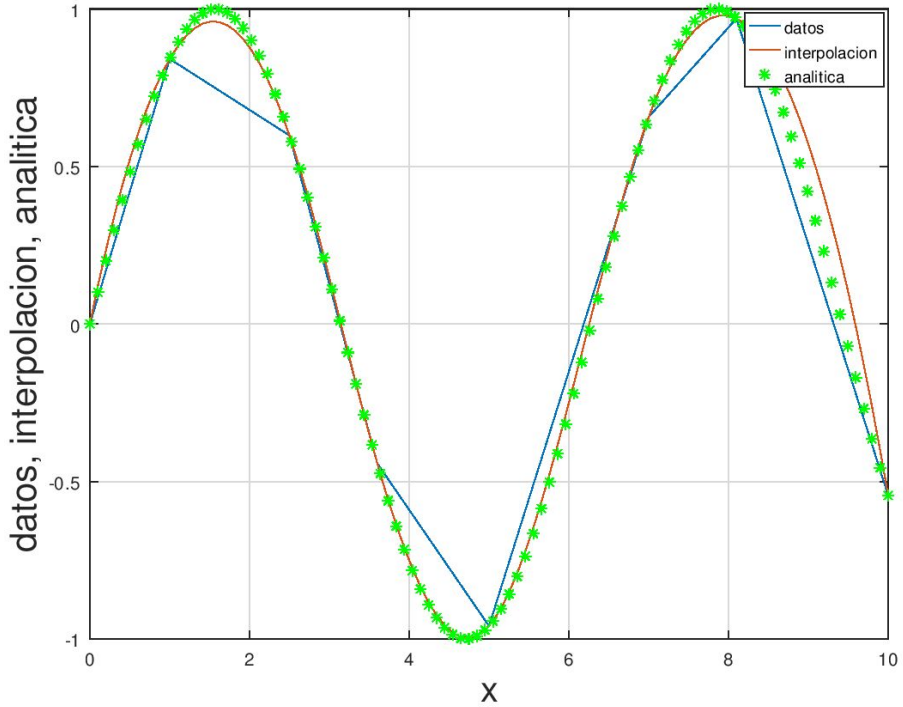
Sintaxis

```
s = spline(x,y,xq)
pp = spline(x,y)
```

Ejemplo

Utilice spline para interpolar una curva sinusoidal sobre puntos de muestra espaciados irregularmente.

```
x = [0 1 2.5 3.6 5 7 8.1 10];
y = sin(x);
xx = 0:.25:10;
yy = spline(x,y,xx);
plot(x,y,'o',xx,yy)
```

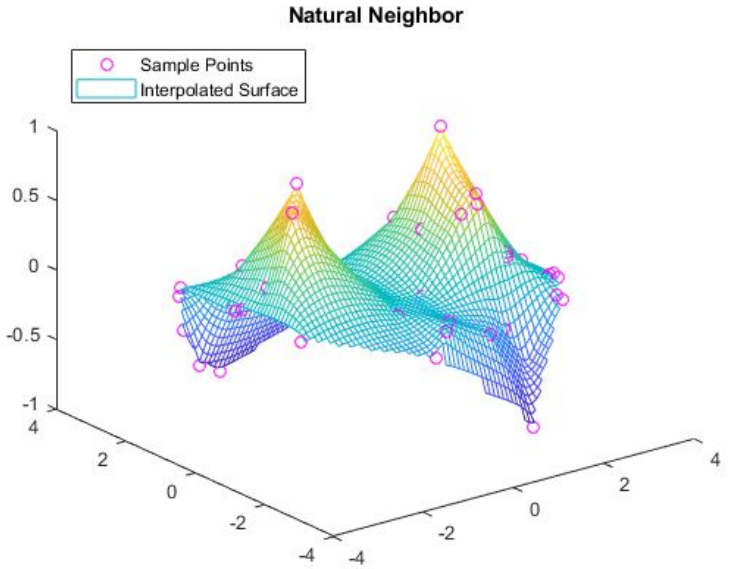


Ejercicio

Utilice spline para interpolar una curva cosenoidal sobre puntos de muestra espaciados irregularmente (los mismos del ejercicio pasado).

Interpolación

Cabe aclarar que se pueden interpolar superficies tridimensionales de una forma “simple” en MATLAB. Sin embargo el ejemplo ilustrativo y el concepto están fuera del alcance del curso. PERO SE PUEDE.



¿Por qué es importante la interpolación?

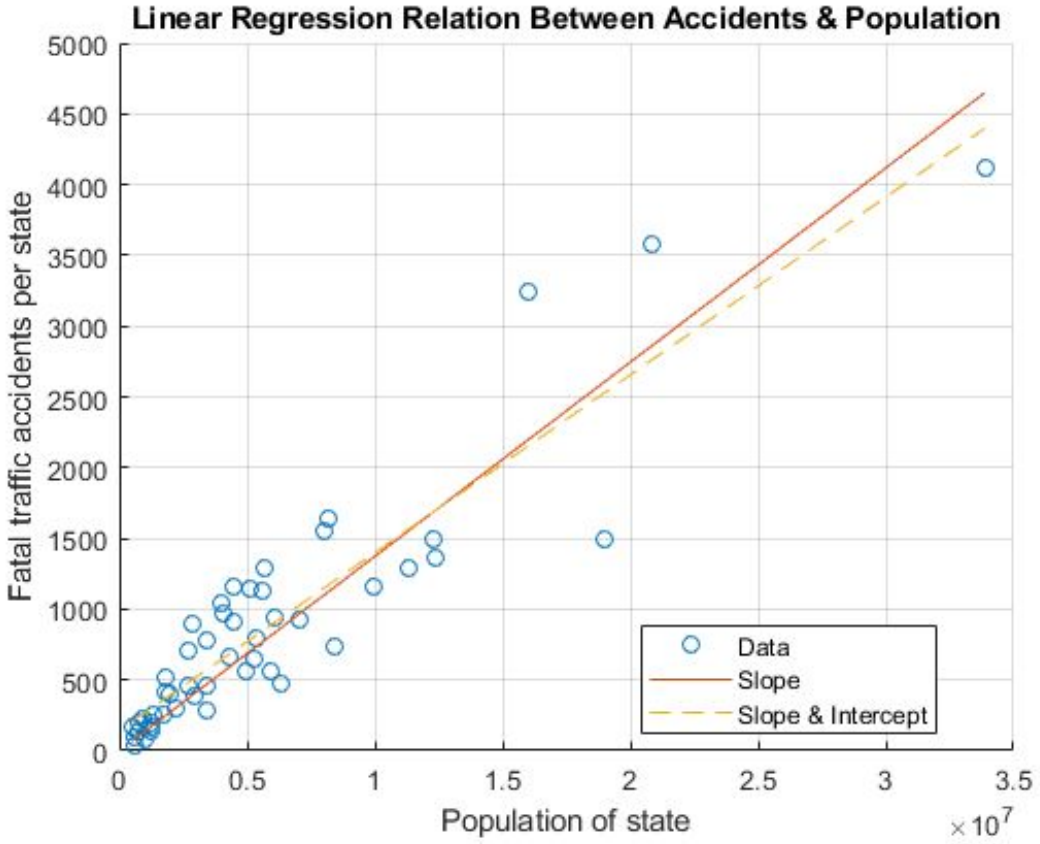
https://en.wikipedia.org/wiki/B%C3%A9zier_curve - **Bézier curve**

Regresión Lineal

El método de regresión es una herramienta muy poderosa en ciencias e ingeniería, ya que permite ajustar un polinomio a un conjunto de datos empíricos, con el fin de obtener información útil dada de forma indirecta por los coeficientes del polinomio. El caso más simple es el caso de regresión lineal.

Puede ajustarse un conjunto de datos que siga una “tendencia” lineal a una línea recta y así calcular los miembros del polinomio lineal (pendiente e intercepto).

Ejemplo


Linear Regression Relation Between Accidents & Population
Fatal traffic accidents per state
Population of state
×10^7
Data
Slope
Slope & Intercept
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
0
0.5
1
1.5
2
2.5
3
3.5

Regresión Lineal en MATLAB

Para efectuar regresión lineal en MATLAB, basta con usar el comando **>>polyfit(argumentos)**

Los argumentos de la función son: **x, y, n.** Por lo tanto, escribimos **polyfit(x, y, n)**

Donde n representa el grado del polinomio que queremos ajustar.

El comando polyfit retorna un arreglo con los coeficientes del polinomio p = [p(1) p(2)] . Para n = 1 (y = p(1)*x + p(2)).

El comando polyval(p, x) evalúa el polinomio en el vector x y retorna un arreglo con esta operación.

Regresión lineal en MATLAB

Utilice `polyfit` para calcular una regresión lineal que predice `y` de `x`:

```
p = polyfit(x,y,1)  p =     1.5229   -2.1911
```

`p(1)` es la pendiente y `p(2)` es la interceptación del predictor lineal.

Llame a `polyval` para usar `p` para predecir `y`, llamando al resultado `yfit`:

```
yfit = polyval(p,x);
```

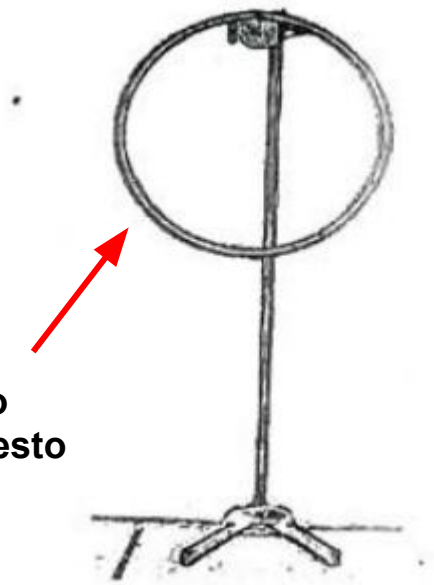
El uso de `polyval` le permite escribir la ecuación de ajuste usted mismo, que en este caso se ve como:

```
yfit =  p(1) * x + p(2);
```

**** Pueden graficar x contra y, y x contra yfit. Para que vean qué tan bien se ajustan los datos con el polinomio del orden seleccionado.**

Ejemplo con datos reales

El período de un péndulo compuesto viene dado por la expresión:



Péndulo Compuesto

$$T = 2\pi\sqrt{\frac{d}{g}}$$

donde T: período.
d: diámetro del anillo.
g: aceleración de la gravedad.

Ejemplo con datos reales

ANILLO	DIÁMETRO MEDIO D (CM)
1	3.84
2	8.5
3	14.75
4	20.85
5	25.85
6	33.2

Diámetro medio de los anillos

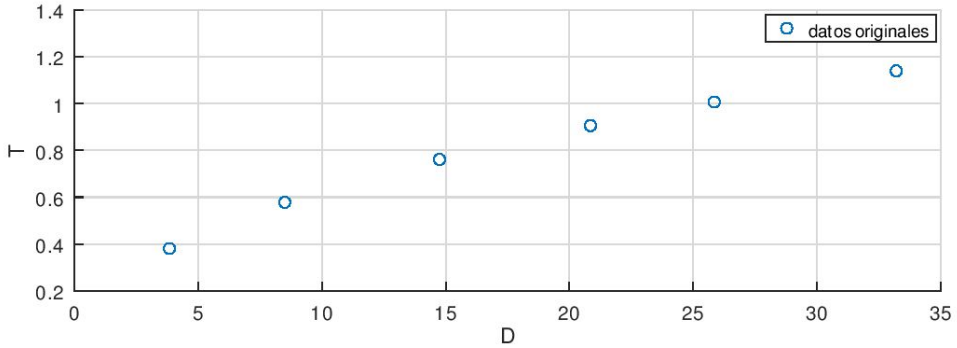
ANILLO	T=t/N
1	0.3808
2	0.57824
3	0.76128
4	0.90648
5	1.00696
6	1.13944

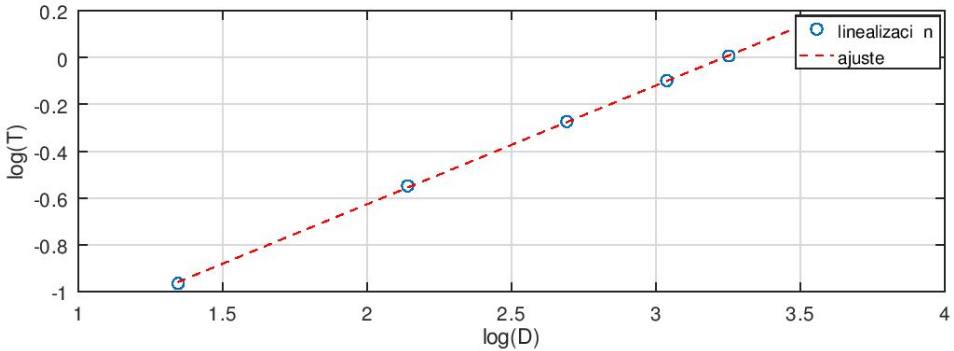
Período de oscilación

Ejemplo con datos reales

```
diametro = [3.84, 8.5, 14.75, 20.85, 25.85, 33.2];
periodo = [0.3808, 0.57824, 0.76128, 0.90648, 1.00696, 1.13944];
subplot(2, 1, 1);
scatter(diametro, periodo);
legend("datos originales");
xlabel("D");
ylabel("T");
grid on;

% Al realizar scatter de los datos
% Nos damos cuenta que se aproxima a una "línea recta".

diam_log = log(diametro);
periodo_log = log(periodo);

%usamos polyval, con n = 1 (linea recta)
p = polyfit(diam_log, periodo_log, 1);
disp(p(1))
disp(p(2))
yfit = polyval(p, diam_log);

subplot(2, 1, 2);
plot(diam_log, periodo_log, "o", diam_log, yfit, "r--");
xlabel("log(D)");
ylabel("log(T)");
legend("linealización", "ajuste");
grid on;
```

Ejemplo con datos reales





m = 0.50746
b = -1.6417

Ejercicio

Realice una regresión lineal del siguiente conjunto de datos. Grafique los datos de la tabla y la regresión lineal en la misma figura.

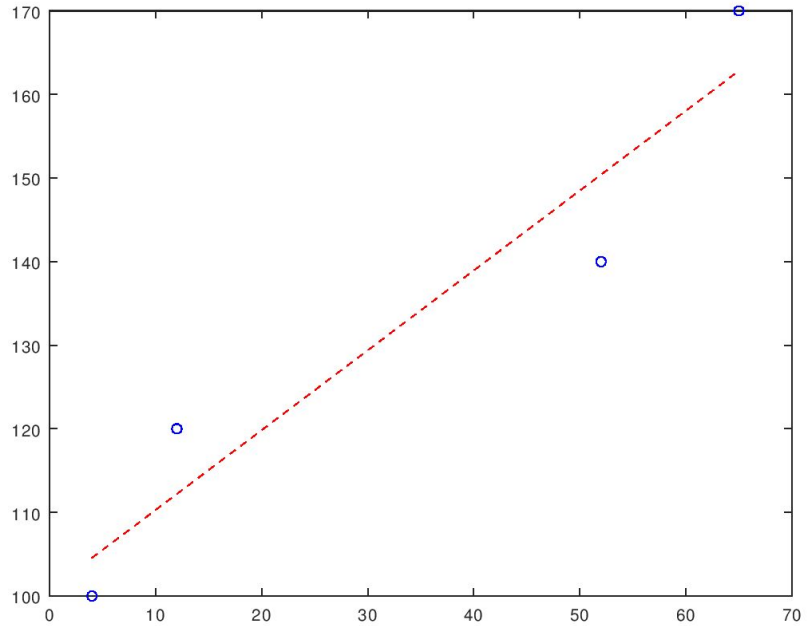
x	y
4	100
12	120
52	140
65	170

R./ m = p(1) = 0.95528

b = p(2) = 100.74

Solución

```
>> x = [4 12 52 65];
>> y = [100, 120, 140, 170];
>> plot(x, y)
>> polyfit(x, y, 1);
>> p = polyfit(x, y, 1);
>> yfit = polyval(p, x);
>> plot(x, y, "bo", x, yfit, "r--");
>> p(1)
ans =  0.95528
>> p(2)
ans =  100.74
```



¿Cuál es la diferencia entre interpolación y regresión?

Con interpolación se ajusta una línea que pase EXACTAMENTE por cada uno de los datos. Con regresión (lineal) se busca un polinomio que reduzca una función de costo. Es decir, la línea que mejor se acomode a un conjunto de datos, sin que necesariamente tome los MISMOS valores del conjunto de datos original.

Se puede hacer regresión con un polinomio de orden n, pero eso lo verán después.