

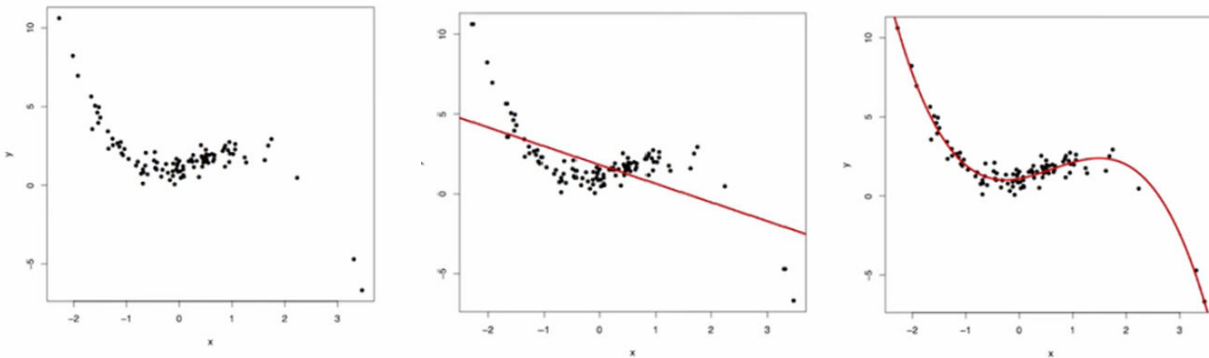
# Regressão polinomial

Nota 1,3

Nessa seção vamos recordar os conceitos de regressão polinomial vistos em sala de aula e revendo as regressões polinomiais de ordem 2 e de ordem 3 nos Exemplos 1 e 2.

Também vamos aprender a usar a função anônima do Matlab e usá-la para definir a equação de regressão e fazer previsões.

Muitas vezes os dados não podem ser ajustados a uma reta (regressão linear). Uma maneira simples de ajustar os dados a uma curva mais elaborada é realizar uma regressão polinomial. Veja o seguinte exemplo:



Na figura central temos o ajuste de uma reta através de regressão linear. Na figura mais a direita o ajuste de uma curva polinomial.

O que é uma curva polinomial? É uma curva onde estimamos o valor da variável dependente (Y) a partir de uma equação envolvendo varias potências da variável X.

Equação de um polinômio de ordem  $K$ :  $\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2 + \dots + \hat{\beta}_K X^K$

Para calcular a regressão de ordem 1, usamos os dados e as equações da regressão linear.

Para carcular uma regressão linear de ordem 2, incluímos uma coluna com os valores de X elevados ao quadrado, na matriz X.

Para carcular uma regressão linear de ordem 3, incluímos uma coluna com os valores de X elevados ao cubo, na matriz X.

E assim sucessivamente.

## Exemplo 1: Regressão Polinomial de Ordem 2

Vamos realizar a regressão polinomial de ordem 2 do exemplo 1 nos slides.

Vamos realizar a regressão polinomial de ordem 2 do exemplo 1 nos slides.

Ler o arquivo Exemplo1.csv para a tabela T.

Salvar a coluna T.Y na variável Y

Criar a variável X0 com valores 1

Salvar a coluna T.X na variável X1

Salvar o quadrado dos valores da coluna X1 na variável X2

Concatenar as variáveis X0, X1 e X2 para forma a matriz X.

```
T = readtable("Exemplo1.csv");
Y = T.Y;
X0 = ones(size(T,1),1);
X1 = T.X;
X2 = X1.^2;
X = [X0 X1 X2];
```

### Calcular coeficientes de regressão e estatísticas

Para calcular a regressão usamos o método dos mínimos quadrados (o mesmo algoritmo para regressão linear) que nos fornecerão os coeficientes da equação desejada:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$$

```
% Calcular os coeficientes de regressão e coeficiente de explicação
[b,~,~,~,stats] = regress(Y,X);
b0 = b(1);
b1 = b(2);
b2 = b(3);
fprintf("Beta_0 = %.2f, Beta_1 = %.2f, Beta_2 = %.2f\n", b0, b1, b2);
```

```
Beta_0 = 4.50, Beta_1 = -3.88, Beta_2 = 3.84
```

### Definir e plotar a curva de regressão

A equação encontrada é  $\hat{Y} = 4.5 - 3.875X + 3.8393X^2$

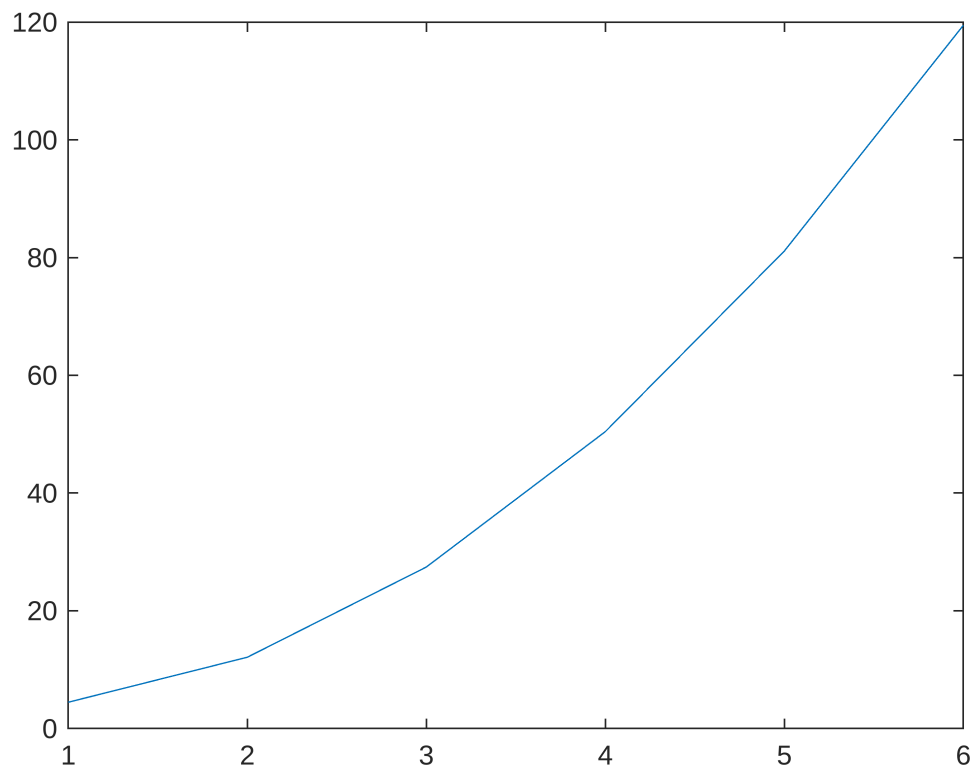
Utilizar a função anônima do Mtlab para definir a equação de regressão

A função anônima é definida pelo operador @(x) seguido da expressão que calcula os valores da função anônima em função de x

Utilizar a função plot para plotar a curva de regressão

```
% Definir a equação de regressão
Yprevisto=@(x)b0 + b1*x + b2*x.^2;

% Plotar a curva de regressão
Yp = Yprevisto(X1);
plot(X1,Yp);
```



## Avaliar $R^2$

Para uma boa qualidade de previsão  $R^2$  deve ser maior ou igual a 0.9

$R^2$  está na segunda posição do vetor stats, calculado pela função regress.

```
R2 = stats(1);
fprintf("Coeficiente de explicação = %.4f\n", R2);
```

Coeficiente de explicação = 0.9987

## Utilizar a equação de regressão para prever valores

Podemos usar a equação de regressão para prever valores.

Podemos observar que os valores 4.5, 5.5, 6.5 e 7.0 não existem no dataset original.

Salvar os valores desejados no vetor Xnovo

Calcular os valores previstos em Yprevisto

```
Xnovo = [4.5 5.5 6.5 7];
Yprevisto(Xnovo)
```

ans = 1×4

## Exemplo 2: Regressão Polinomial de Ordem 3

Vamos realizar a regressão polinomial de ordem 3 do exemplo 2 dos slides.

Ler o arquivo Exemplo2.csv para a tabela T.

Salvar a coluna T.Y na variável Y

Criar a variável X0 com valores 1

Salvar a coluna T.X na variável X1

Salvar o quadrado dos valores da coluna X1 na variável X2

Salvar o cubo dos valores da coluna X1 na variável X3

Concatenar as variáveis X0 X1 X2 X3 para formar a matriz X

```
T = readtable( "Exemplo2.csv" )
```

T = 8x2 table

	Y	X
1	5	1
2	10	2
3	30	3
4	35	4
5	20	5
6	40	6
7	80	7
8	120	8

```
Y = T.Y;
X0 = ones(size(T,1),1);
X1 = T.X;
X2 = X1.^2;
X3 = X1.^3;
X = [X0 X1 X2 X3];
```

### Calcular coeficientes de regressão e estatísticas

Para calcular a regressão usamos o método dos mínimos quadrados (o mesmo algoritmo para regressão linear) que nos fornecerão os coeficientes da equação desejada:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2 + \hat{\beta}_3 X^3$$

```
% Calcular os coeficientes e estatísticas
[b,~,~,~,stats] = regress(Y,X);
b0 = b(1);
```

```

b1 = b(2);
b2 = b(3);
b3 = b(4);
R2 = stats(1);
fprintf("Beta_0 = %.2f, Beta_1 = %.2f, Beta_2 = %.2f, Beta_3 = %.2f\n", b0, b1, b2, b3);

```

```

Beta_0 = -31.07, Beta_1 = 44.14, Beta_2 = -11.80, Beta_3 = 1.09

```

```

fprintf("Coeficiente de explicação = %.4f\n", R2);

```

```

Coeficiente de explicação = 0.9642

```

### Definir a equação de regressão

A equação encontrada é  $\hat{Y} = -31.0714 + 44.1360X - 11.8019X^2 + 1.0859X^3$

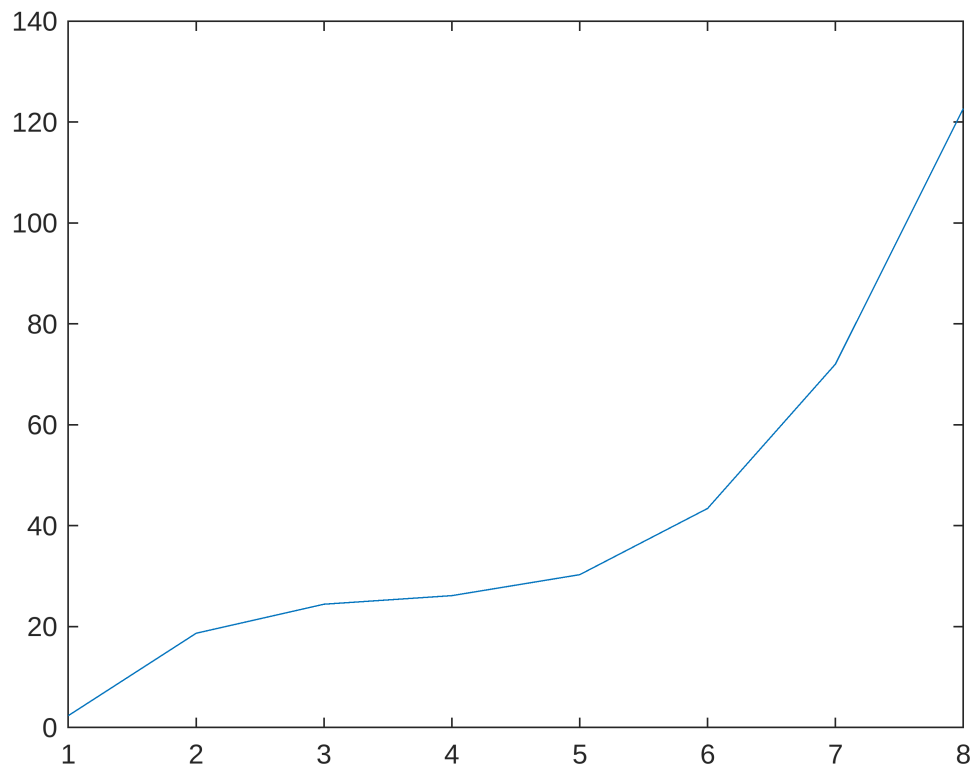
Definir a equação de regressão utilizando a função anônima do MatLab

```

% Definir a equação de regressão
Yprevisto=@(x)b0 + b1*x + b2*x.^2 + b3*x.^3;

% Plotar a curva de regressão
Yp = Yprevisto(X1);
plot(X1,Yp);

```



### Utilizar a equação de regressão para prever valores

Podemos usar a equação de regressão para prever valores.

Podemos observar que os valores 4.5, 5.5, 6.5 e 7.0 não existem no dataset original.

Salvar os valores desejados no vetor Xnovo

Calcular os valores previstos em Yprevisto

```
Xnovo = [4.5 5.5 6.5 7.0];  
Yprevisto(Xnovo)
```

```
ans = 1x4  
    27.5000    35.3274    55.3842    72.0346
```

## TDE02

### Regressão usando arquivo de dados de projeto

No arquivo EsforçoProjetos3 temos armazenados os valores de estimativas de tamanho do projeto (TAMANHO) em pontos de função e a produtividade obtida no desenvolvimento (PRODUTIVIDADE) medida pela quantidade de pontos de função desenvolvidos por hora. Queremos construir um modelo para previsão da produtividade baseada em regressão polinomial.

Nesse TDE você deve testar o modelo de regressão linear simples, regressão polinomial de segundo e regressão linear do terceiro grau.

No final deve selecionar o modelo mais adequado e fazer algumas previsões.

Os códigos para regressão linear e regressão polinomial de ordem 2 já estão prontos.

### Ler o arquivo de projetos e preparar as variáveis

Carregar a coluna PRODUTIVIDADE na variável Y

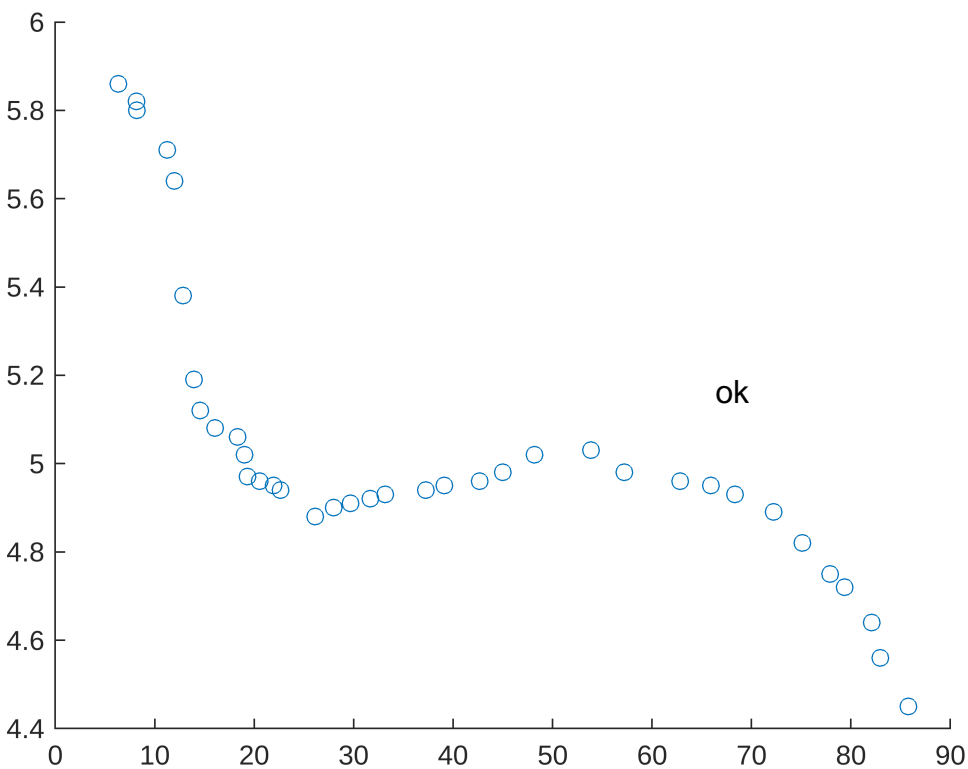
Carregar a coluna TAMANHO na variável X1

Calcular a variável  $X_2 = (X_1)^2$  (a exponenciação é realizada elemento a elemento)

Calcular a variável  $X_3 = (X_1)^3$  (a exponenciação é realizada elemento a elemento)

Plotar o diagrama de dispersão de X1 em relação a X2.

```
T = readtable("EsforcoProjetos3.csv");  
Y = T.PRODUTIVIDADE;  
X0 = ones(size(T,1),1);  
X1 = T.TAMANHO;  
X2 = X1.^2;  
X3 = X1.^3;  
scatter(X1, Y)
```



## Regressão linear

### Calcular coeficientes de regressão e estatísticas

Para calcular a regressão usamos o método dos mínimos quadrados (o mesmo algoritmo para regressão linear) que nos fornecerão os coeficientes da equação desejada:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X$$

Na regressão linear a matriz X terá apenas os valores de X0 e X1

```
% Calcular a matriz X
X = [X0 X1];

% Calcular os coeficientes de regressão e coeficiente de explicação
[b,~,~,~,stats] = regress(Y,X);
b0 = b(1);
b1 = b(2);
R2 = stats(1);
fprintf("Beta_0 = %.4f, Beta_1 = %.4f\n", b0, b1);
```

```
Beta_0 = 5.4117, Beta_1 = -0.0094
```

```
fprintf("Coeficiente de explicação = %.4f\n", R2);
```

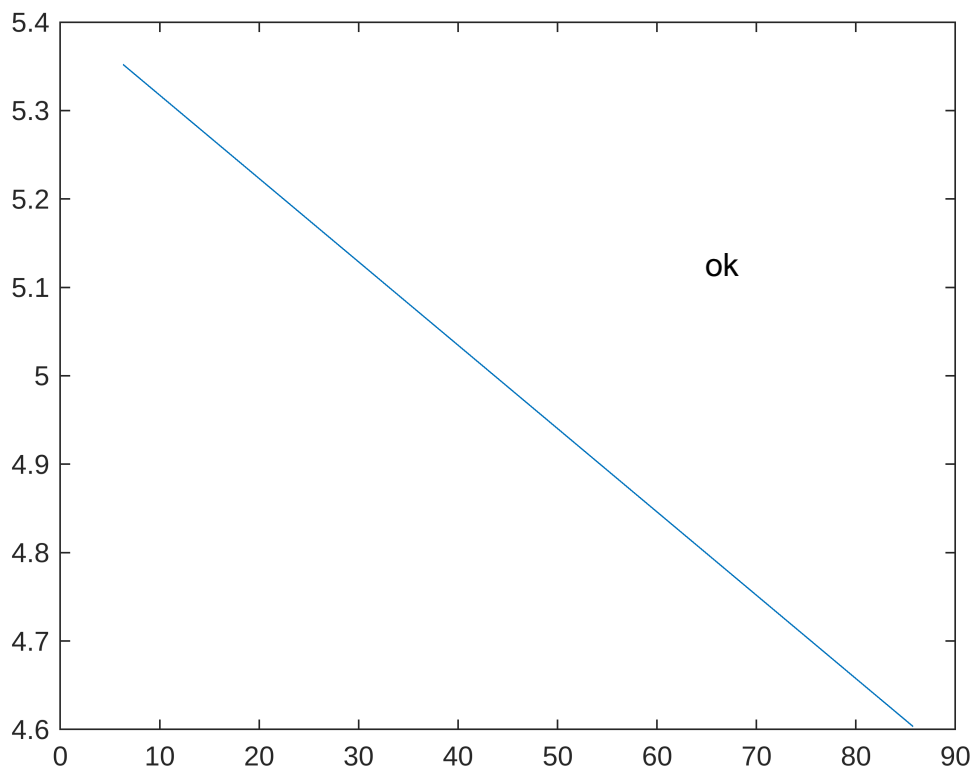
```
Coeficiente de explicação = 0.5315
```

## Definir a equação de regressão e plotar a reta de regressão

Definir a equação de regressão utilizando a função anônima do MatLab

```
% Definir a equação de regressão
Yprevisto=@(x)b0 + b1*x;

% Plotar a curva de regressão
Yp = Yprevisto(X1);
plot(X1,Yp);
```



## Regressão polinomial de ordem 2

Para calcular a regressão usamos o método dos mínimos quadrados (o mesmo algoritmo para regressão linear) que nos fornecerão os coeficientes da equação desejada:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2$$

Na regressão polinomial de ordem 2 a matriz X terá os valores de X0 X1 e X2

```
% Calcular a matriz X
X = [X0 X1 X2];

% Calcular os parâmetros, os coeficientes F e e R2
```



```
[b,~,~,~,stats] = regress(Y,X);
b0 = b(1);
b1 = b(2);
b2 = b(3);
R2 = stats(1);
fprintf("Beta_0 = %.4f, Beta_1 = %.4f, Beta_2 = %.4f \n", b0, b1, b2);
```

```
Beta_0 = 5.6732, Beta_1 = -0.0267, Beta_2 = 0.0002
```

```
fprintf("Coeficiente de explicação = %.4f\n", R2);
```

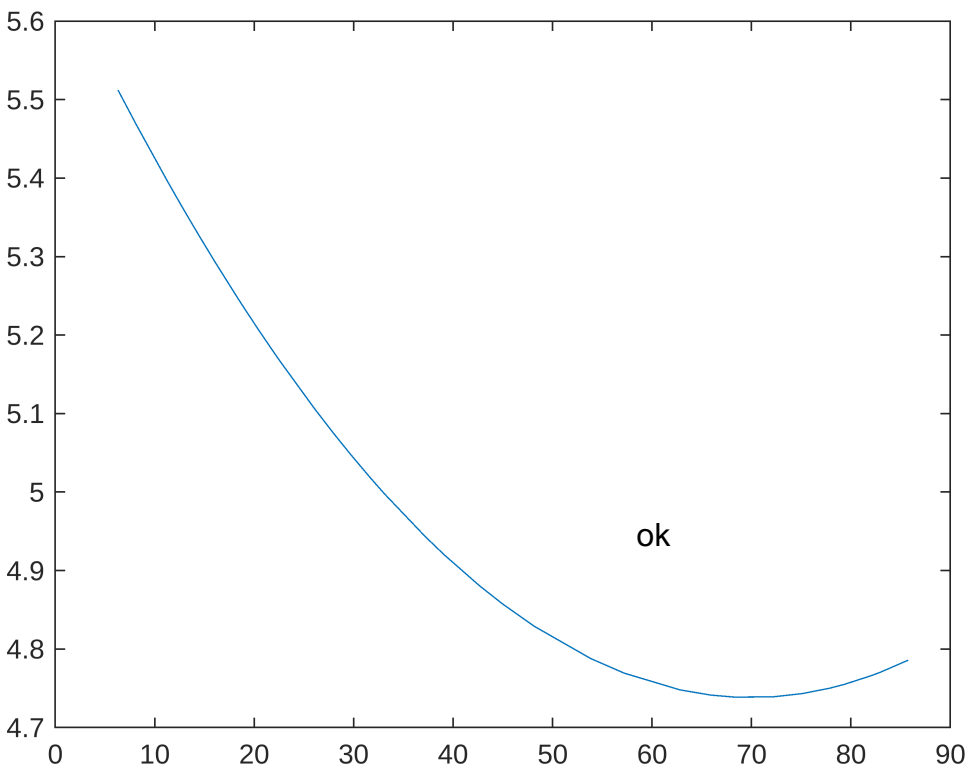
```
Coeficiente de explicação = 0.6122
```

## Definir a equação de regressão e plotar a curva de regressão

Definir a equação de regressão utilizando a função anônima do MatLab

```
% Definir a equação de regressão
Yprevisto=@(x)b0 + b1*x + b2*x.^2;

% Plotar a curva de regressão
Yp = Yprevisto(X1);
plot(X1,Yp);
```



## TDE02 - Regressão polinomial de ordem 3

## Calcular coeficientes de regressão e estatísticas

Para calcular a regressão usamos o método dos mínimos quadrados (o mesmo algoritmo para regressão linear) que nos fornecerão os coeficientes da equação desejada:

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\beta}_2 X^2 + \hat{\beta}_3 X^3$$

Na regressão polinomial de ordem 2 a matriz X terá os valores de X0 X1 X2 e X3

```
% Colocar o seu código aqui
% Calcular a matriz X
X = [X0, X1, X2, X3];
% Calcular os parâmetros, os coeficientes F e e R2
[b,~,~,~,stats] = regress(Y,X);
b0 = b(1);
b1 = b(2);
b2 = b(3);
b3 = b(4);
R2 = stats(1);

fprintf("Beta_0 = %.6f, Beta_1 = %.6f, Beta_2 = %.6f, Beta_3 = %.6f\n", b0, b1, b2, b3);

Beta_0 = 6.518991, Beta_1 = -0.118679, Beta_2 = 0.002669, Beta_3 = -0.000018

fprintf("Coeficiente de explicação = %.4f\n", R2);

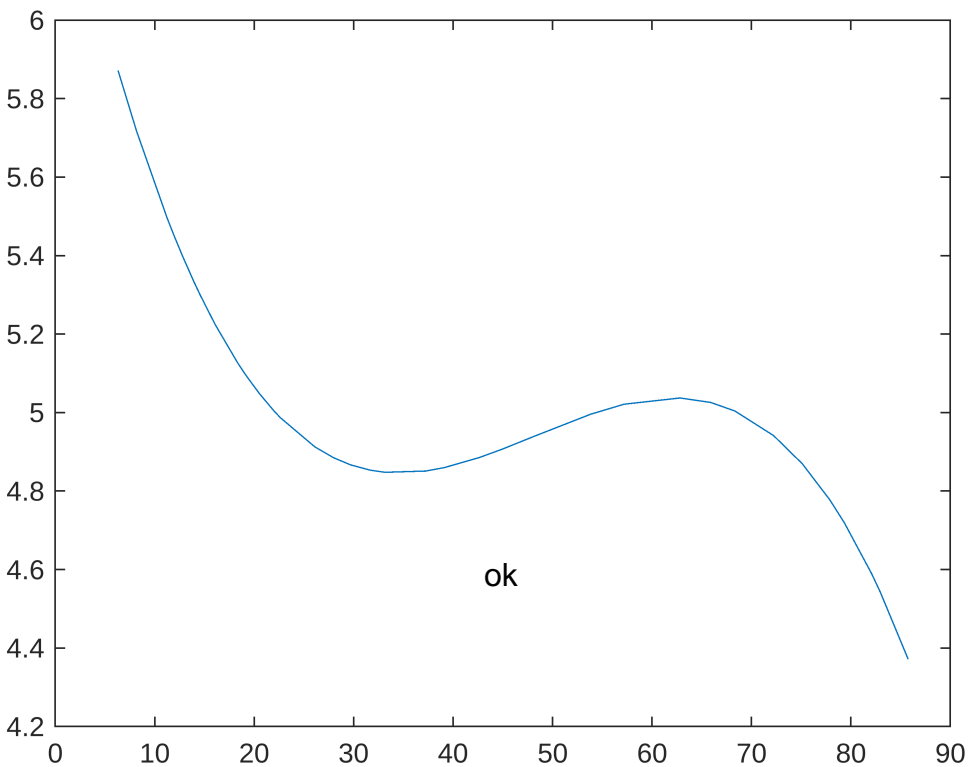
Coeficiente de explicação = 0.9259
```

## Definir a equação de regressão e plotar a curva de regressão

Definir a equação de regressão utilizando a função anônima do MatLab

```
% Colocar seu código aqui
% Definir a equação de regressão
Yprevisto=@(x)b0 + b1*x + b2*x.^2 + b3*x.^3;

% Plotar a curva de regressão
Yp = Yprevisto(X1);
plot(X1,Yp);
```



## Utilizar a melhor equação de regressão para prever valores

Selecionar o modelo de regressão que tiver o maior valor de R2

```
% Colocar seu código aqui
% Calcular a matriz X (o mesmo cálculo do melhor modelo)
X = [X0, X1, X2, X3];
% Calcular os coeficientes e estatísticas
[b,~,~,~,stats] = regress(Y,X);
b0 = b(1);
b1 = b(2);
b2 = b(3);
b3 = b(4);
R2 = stats(1);
% Definir a equação de regressão
Yprevisto=@(x)b0 + b1*x + b2*x.^2 + b3*x.^3;
```

Fazer as previsões para os seguintes valores em Xnovo

```
Xnovo = [5.81 9.15 15.06 17.00 23.42 32.11 49.5 56.82 76.20 87.25]
```

```
Xnovo = 1x10
    5.8100    9.1500   15.0600   17.0000   23.4200   32.1100   49.5000   56.8200 ...
```

Calcular os valores previstos em Yprevisto

```
Ypprevisto = Yprevisto(Xnovo)
```

```
Ypprevisto = 1×10  
5.9160    5.6424    5.2742    5.1824    4.9672    4.8512    4.9535    5.0189 ...
```

Corret

## Entrega

Resolver a regressão linear de ordem 3.

Executar o live script com a opção VIEW Output Inline.

Exportar para pdf: ABA LIVE EDITOR, OPÇÃO Save->Export to PDF

Fazer o upload do PDF.

Trabalho em grupo. Apenas o líder deve fazer o upload.