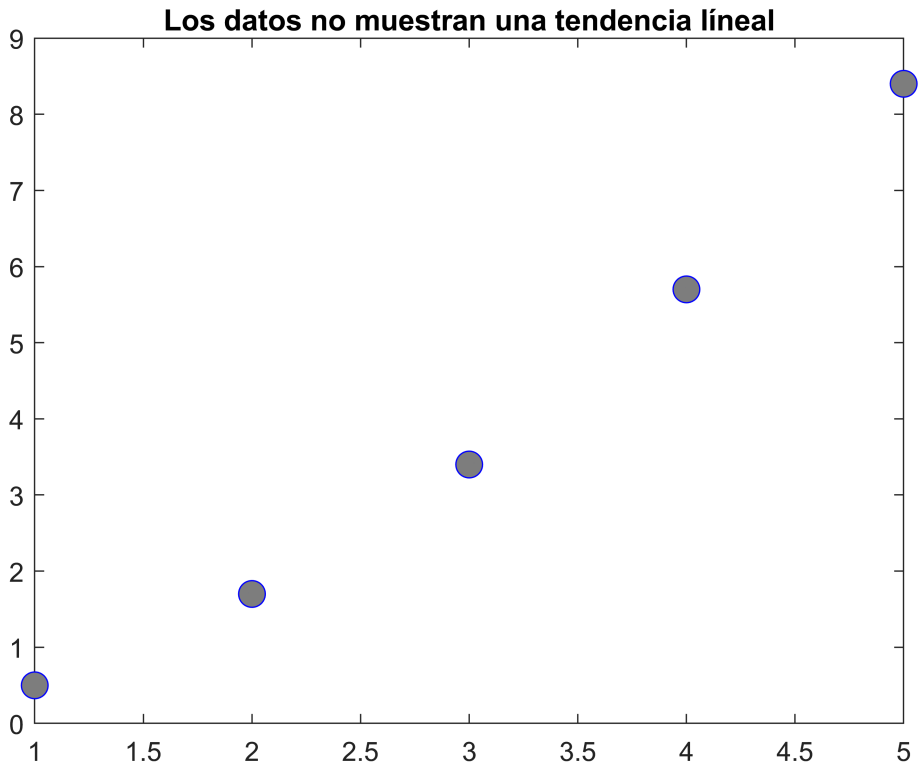
```
clf
clear all
x=1:5
```

```
x = 1×5
     1     2     3     4     5
```

```
y=[0.5 1.7 3.4 5.7 8.4]
```

```
y = 1×5
    0.5000    1.7000    3.4000    5.7000    8.4000
```

```
plot(x,y,'o','MarkerSize',10,'MarkerEdgeColor','b','MarkerFaceColor',[0.5,0.5,0.5])
title('Los datos no muestran una tendencia líneal')
```



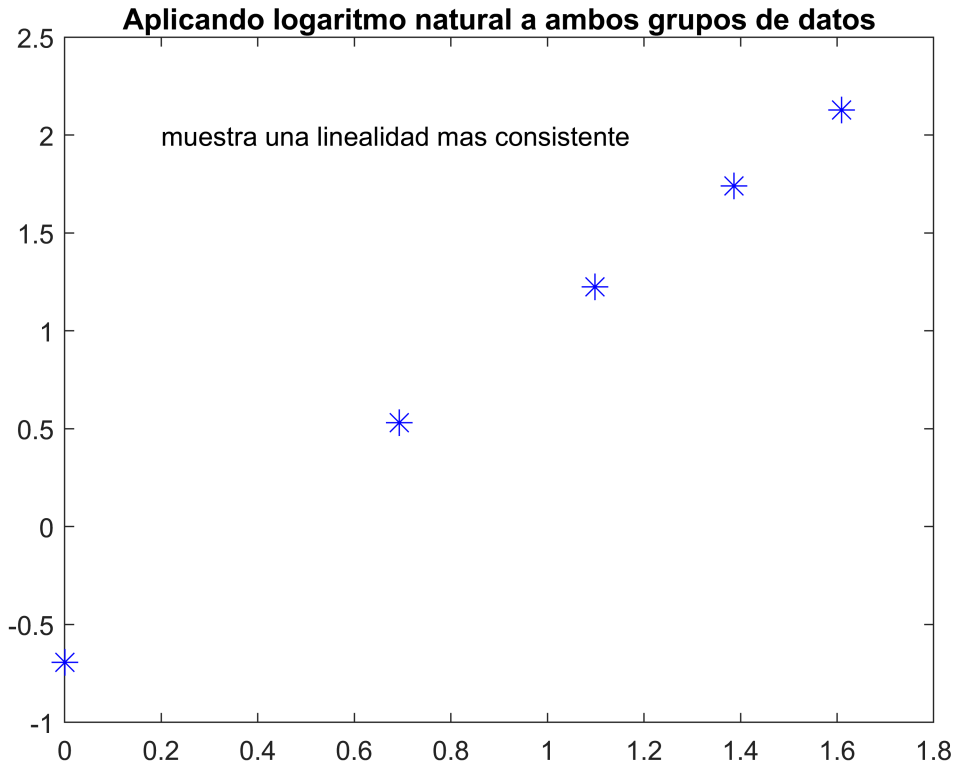

```
clf
X=log(x)
```

```
X = 1×5
         0    0.6931    1.0986    1.3863    1.6094
```

```
Y=log(y)
```

```
Y = 1×5
   -0.6931    0.5306    1.2238    1.7405    2.1282
```

```
plot(X,Y,'*','MarkerSize',10,'MarkerEdgeColor','b','MarkerFaceColor',[0.5,0.5,0.5])
title('Aplicando logaritmo natural a ambos grupos de datos')
text(0.2,2,'muestra una linealidad mas consistente')
```




La función de potencia es la adecuada para representar estos datos

$$y = b\,x^m$$
$$\ln y = \ln b + \ln x^m$$
$$\ln y = \ln b + m \ln x$$
$$Y = B + m\,X$$

ln (y) es Y & ln (x) es X, como en la transformación realizada en la grafica anterior.

Con los datos transformados podemos aplicar el modelo lineal

```
n=5
```

```
n = 5
```

```
Sx=sum(X)
```

```
Sx = 4.7875
```

```
Sy=sum(Y)
```

```
Sy = 4.9300
```

```
Sx2=sum(X.^2)
```

```
Sx2 = 6.1995
```

```
Sxy=sum(X.*Y)
```

```
Sxy = 7.5503
```

```
A=[n Sx Sy;
    Sx Sx2 Sxy]
```

```
A = 2×3
    5.0000    4.7875    4.9300
    4.7875    6.1995    7.5503
```

```
A(1,:)=A(1,:)/A(1,1);
A(2,:)=A(2,:)-A(1,:)*A(2,1)
```

```
A = 2×3
    1.0000    0.9575    0.9860
         0    1.6155    2.8299
```

```
A(2,:)=A(2,:)/A(2,2);
A(1,:)=A(1,:)-A(2,:)*A(1,2)
```

```
A = 2×3
    1.0000         0   -0.6913
         0    1.0000    1.7517
```

```
a0=A(1,3)
```

```
a0 = -0.6913
```

```
a1=A(2,3)
```

```
a1 = 1.7517
```

los coeficientes anteriores corresponden al modelo lineal de los datos transformados.

$Y = B + m\,X$

ahora se requieren para ser usados en el modelo de potencia

$b = \exp B$

$y = b\,x^m$

```
m=a1
```

```
m = 1.7517
```

```
B=a0
```

```
B = -0.6913
```

```
b=exp(B)
```

```
b = 0.5009
```

```
fp=@(X) b*X.^m
```

```
fp = function_handle with value:
```

```
    @(X)b*X.^m
```

```
equis=min(x):0.1:max(x)
```

```
equis = 1×41
    1.0000    1.1000    1.2000    1.3000    1.4000    1.5000    1.6000    1.7000 ···
```

```
yes=fp(equis)
```

```
yes = 1×41
    0.5009    0.5920    0.6894    0.7932    0.9031    1.0192    1.1411    1.2690 ···
```

```
clf
plot(equis,yes)
x=1:5
```

```
x = 1×5
     1     2     3     4     5
```

```
y=[0.5 1.7 3.4 5.7 8.4]
```

```
y = 1×5
    0.5000    1.7000    3.4000    5.7000    8.4000
```

```
hold on
plot(x,y,'*')
text(1.25,6,'conjunto de datos originales y el modelo de la función potencia')
```