```
%% Taller
clc
clear all

%% Item 1-A
% Se definen 2 instrumentos lineales que representan medidas de variables:
% Para este ejemplo crearemos dos vectores X1 y X2 que seran tomadas en
% cuenta como medidas de la cantidad de particulado suspendido por unidad de
% volumen de Aire [ug/m^3], adquiridas por un sensor laser de concentración
% de partículas (PDS - Particle Density Sensor).

% Comunmente se encuentra en el mercado sensores de este tipo que permiten
% obtener medidas para diferentes rangos de tamaño de particulas entre 0.3um y 10um
% pero para el presente fin solo hara falta un rango arbitrario.

% Supondremos que disponemos de medidas para particulas con tamaños entre 1.0um y 2.5↙
um
% Ademas cabe aclarar que los PDS retornan la informacion medida como una señal↙
digital,
% por ejemplo de 16 bits (valor entre 0 y 2^16 -1), para este ejercicio es oportuno
% suponer que la señal es analogica y esta comprendida entre 0v-5v o 0v-3.3v .

% El rango de medida de nuestra sensor X1 estara comprendido entre 0-999 ug/m^3
% con una resolucion de 0.015 ug/m^3 tomando como referencia a un sensor
% comercial (HK-A5) https://www.components-mart.pl/datasheets/3f/SEN0177.pdf
% Rango completo 0-999 ug/m^3 tomando el mayor numero de medidas diferentes
% otorgado por su resolucion.

% Para el vector de datos X2 cambiaremos las caracteristicas de nuestro
% sensor con fines practicos.

X1= 0:0.015:999;
X2= 0:0.03:500;

% Cálculo de la respuesta ideal de cada sistema
% Para esto definimos nuestros sistemas con deriva cero es decir B1=B2=0
% Y una sensibilidad de 5/999 y 3.3/500 dada en [v/(ug/m^3)] para cada
% sistema respectivamente.

S1 = 5/999;
S2 = 3.3/500;
B1 = 0;
B2 = 0;

Y1 = S1*X1 + B1;
Y2 = S2*X2 + B2;

% Dibujar las respuestas ideales
figure('rend','painters','pos',[50 50 900 600])
subplot(2,1,1);
plot(X1,Y1,'Color','r');
title('Sistema 1');
ylabel('Y1 [V]')
xlabel('X1 [ug/m^3]')
axis([0 max(X1) 0 5.1]);
```

```
grid on;

subplot(2,1,2);
plot(X2,Y2,'Color','b');
title('Sistema 2');
ylabel('Y2 [V]')
xlabel('X2 [ug/m^3]')
axis([0 max(X2) 0 3.4]);
grid on;

% Submuestreamos los vectores X1 y X2 para obtener vectores de datos, mas
% acordes a lo que en la realidad seria una toma de datos

X1ss = downsample(X1,300);
X2ss = downsample(X2,150);

% Ahora simularemos un ruido que afecte la deriva de nuestros sistemas
% Estos son los ruidos Rb1 y Rb2 con unidades de [V]

Rb1 = 0.09 + 0.14*randn(size(X1ss));
Rb2 = 0.07 + 0.15*randn(size(X2ss));

Y1r =  S1*X1ss + B1 + Rb1;
Y2r =  S2*X2ss + B2 + Rb2;

% Dibujar las respuestas con ruido que afecta la deriva
figure('rend','painters','pos',[50 50 900 600])
subplot(2,1,1);
plot(X1ss,Y1r,'.','Color','r');
hold on;
plot(X1,Y1,'Color','k');
title('Sistema 1 + Ruido de deriva');
ylabel('Y1r [V]')
xlabel('X1 [ug/m^3]')
legend('Y1 con Rb1','Y1 ideal')
axis([0 max(X1ss) 0 5.1]);
grid on;

subplot(2,1,2);
plot(X2ss,Y2r,'.','Color','b');
hold on;
plot(X2,Y2,'Color','k');
title('Sistema 2 + Ruido de deriva');
ylabel('Y2r [V]')
xlabel('X2 [ug/m^3]')
legend('Y2 con Rb2','Y2 ideal')
axis([0 max(X2ss) 0 3.4]);
grid on;


% Ahora simularemos un ruido que afecte la sensibilidad de nuestros sistemas
% Estos son los ruidos Rs1 y Rs2 con unidades de [V/(ug/m^3)]

Rs1 = 0.000018 + 0.00007*randn(size(X1ss));
Rs2 = 0.0009 + 0.0006*randn(size(X2ss));
```

```
Y1rs =  (S1 + Rs1).*X1ss + B1;
Y2rs =  (S2 + Rs2).*X2ss + B2;

% Dibujar las respuestas con ruido que afecta la sensibilidad
figure('rend','painters','pos',[50 50 900 600])
subplot(2,1,1);
plot(X1ss,Y1rs,'.','Color','r');
hold on;
plot(X1,Y1,'Color','k');
title('Sistema 1 + Ruido en la sensibilidad');
ylabel('Y1rs [V]')
xlabel('X1 [ug/m^3]')
legend('Y1 con Rs1','Y1 ideal')
axis([0 max(X1ss) 0 max(Y1rs)]);
grid on;

subplot(2,1,2);
plot(X2ss,Y2rs,'.','Color','b');
hold on;
plot(X2,Y2,'Color','k');
title('Sistema 2 + Ruido en la sensibilidad');
ylabel('Y2rs [V]')
xlabel('X2 [ug/m^3]')
legend('Y2 con Rs2','Y2 ideal')
axis([0 max(X2ss) 0 max(Y2rs)]);
grid on;

% Por ultimo generamos señales para los sistemas 1 y 2 adicionando ruido en
% la deriva (Rb) y en la sensibilidad (Rs) para observar su efecto.

Y1rbs =  (S1 + Rs1).*X1ss + B1 + Rb1;
Y2rbs =  (S2 + Rs2).*X2ss + B2 + Rb2;

% Dibujar las respuestas con ruido que afecta la sensibilidad
figure('rend','painters','pos',[50 50 900 600])
subplot(2,1,1);
plot(X1ss,Y1rbs,'.','Color','r');
hold on;
plot(X1,Y1,'Color','k');
title('Sistema 1 + Ruidos Rb1 y Rs1');
ylabel('Y1rbs [V]')
xlabel('X1 [ug/m^3]')
legend('Y1 con Rb1 y Rs1','Y1 ideal')
axis([0 max(X1ss) 0 max(Y1rbs)]);
grid on;

subplot(2,1,2);
plot(X2ss,Y2rbs,'.','Color','b');
hold on;
plot(X2,Y2,'Color','k');
title('Sistema 2 + Ruidos Rb2 y Rs2');
ylabel('Y2rbs [V]')
xlabel('X2 [ug/m^3]')
legend('Y2 con Rb2 y Rs2','Y2 ideal')
```

```
axis([0 max(X2ss) 0 max(Y2rbs)]);
grid on;
hold off;

%% Item 1-B
% Seleccionamos un modelo del punto anterior, por ejemplo el Yrbs1 que
% posee los dos tipos de ruido mencionados en el item anterio, y
% procederemos a realizar 20 experimentos (toma de datos) bajo las mismas
% condiciones, para su posterior analisis
figure('rend','painters','pos',[50 50 900 600])

Yexp = zeros(20,size(X1ss,2));
for i=1:1:20
    Rb1 = 0.09 + 0.14*randn(size(X1ss));
    Rs1 = 0.000018 + 0.0003*randn(size(X1ss));
    Yexp(i,:) =  (S1 + Rs1).*X1ss + B1 + Rb1;
    plot(Yexp(i,:),X1ss,'.')
    hold on;
end
hold off
legend('Exp 1', 'Exp 2', 'Exp 3', 'Exp 4', 'Exp 5', 'Exp 6', 'Exp 7', 'Exp 8', 'Exp↙
9','Exp 10','Exp 11', 'Exp 12', 'Exp 13', 'Exp 14', 'Exp 15', 'Exp 16', 'Exp 17', 'Exp↙
18', 'Exp 19','Exp 20')
title('20 Experimentos con el Sistema Yrsb1');
ylabel('Experimentos Y1rbs [V]')
xlabel('X1 [ug/m^3]')
grid on;

% Recuperamos uno de los experimentos para realizar el ajuste de curva con
% la APP de Curve Fitting

Yajuste = Yexp(8,:);

% Con el codigo generado por la App Curve Fitting podemos generar la grafica
% del ajuste de curva polinomial de grado 1 y ademas imprimir la ecuación
% en la ventana de comandos.
disp('Ajuste de Curva Experimento 8 (Polinomial de grado 1)')
curveFitting(X1ss,Yajuste)

% Tambien usamos el codigo generado por la App para el ajuste con la opcion
% Center and Scale
disp('Ajuste de Curva Experimento 8 (Polinomial de grado 1 con Center and Scale)')
curveFittingCenterAndScale(X1ss,Yajuste)

% Al momento de generar el ajuste de curva nos retorna varios resultados
% que nos describen la calidad de la regresion, por este motivo es
% importante que conozcamos su significado.

% SSE :
% R-Square:
% Adjusted R-Square:
% RMSE :

% ¿Que sucede cuando se activa la opcion Center and Scale?
```

```
% Esta opcion suele ser sugerida cuando en el procedimiento se utilizan
% los valores del vector X y un ajuste polinomial de alto orden, o los valores
% del vector X son muy grandes. Esto se debe a que su gran magnitud se propaga
% a traves de los cálculos realizados para la obtencion del ajuste de curva.

% Para solucionar este problema, se normalizan los datos.
% La normalización escala los datos para mejorar la precisión de los cálculos↙
numéricos
% posteriores. Una forma de normalizar X es centrarlo en la media cero y
% escalarlo a la unidad de desviación estándar. El código equivalente es:
% Xnorm = (X-mean(X)) / std (X)

% Esto se hace antes de que se calculen los términos polinomiales por este
% motivo los valores de p1 y p2 se ven afectados. Igualmente para ajustes
% de orden polinomial alto pueden verse afectados los valores de Goodness
% of fit, no significando que su rigor teorico se haya visto comprometido,
% solo se debe a la capacidad que tiene una computadora a la hora de
% realizar operaciones de alta presicion.

% Por este motivo notamos que para este caso en particular no hay diferencia
% en los parametros de calidad del ajuste de curva entre
% los 2 metodos (With Center and Scale , WithOut Center and Scale)

%Fuente:
%https://www.mathworks.com/help/curvefit/fit-comparison-in-curve-fitting-app.html
%Seccion: About Scaling

%% Item 1-C
%% Regresion usando los datos de todos los EXPERIMENTOS
% Para realizar una regresion usando todos los datos de los experimentos
% basta con generar un solo vector de datos X = [Xexp1 Xexp2 Xexp3 ... XexpN]
% y de la misma forma un solo vector Y formado con sus imagenes respectivamente,
%Xreg = zeros(1,20*size(X1ss,2));
for i=1:1:20
   Xreg(1, (i-1)*size(X1ss,2) + 1 : i*size(X1ss,2) ) = X1ss;
   Yreg(1, (i-1)*size(X1ss,2) + 1 : i*size(X1ss,2) ) = Yexp(i,:);
end

disp('Ajuste de Curva con todos los Experimentos')
curveFittingFullExperiments(Xreg, Yreg)
% Fuente:
% https://www.mathworks.com/matlabcentral/answers/334814-simultaneously-fit-multiple-↙
datasets
% Seccion: Respuesta por parte del personal de MathWorks (Vandana Ravichandran)

%% Item 1-D
% Cambiar el modelo lineal por uno no lineal al red**edor de un punto de la
% grafica, para esto se genera una funcion no lineal, para este ejemplo de
% caracter polinomial de orden 3 de tal forma que se cruce con nuestro
% modelo lineal en un punto determinado (centro de la grafica) de la
% siguiente manera:

Rbnl = 0.028 + 0.07*randn(size(X1ss));

Ylineal =  S1*X1ss + B1 + Rbnl;
```

```
Yorden3 =  S1*0.00002*(X1ss-700).^3 + B1 + Rbnl +3.2;

% Dibujar las respuesta
figure('rend','painters','pos',[50 50 900 600])
subplot(2,1,1);
plot(X1ss,Ylineal,'.','Color','r');
hold on;
plot(X1ss,Yorden3,'.','Color','b');
plot(X1,Y1,'Color','k');
title('Sistemas lineal y de orden 3');
ylabel('Y [V]')
xlabel('X [ug/m^3]')
legend('Y Lineal + ruido deriva','Y orden 3','Y1 ideal')
axis([0 max(X1ss) 0 max(Yorden3)]);
grid on;

% Ahora procederemos a generar una funcion a trozos haciendo uso del modelo
% lineal y no lineal, aprovechando el punto de corte anteriormente conseguido.

YnoLineal = [Ylineal(1:floor(size(Ylineal,2)/2)) Yorden3(floor(size(Ylineal,2)/2) + 1↙
: size(Ylineal,2))];

% Generamos la grafica de este ultimo sistema
figure('rend','painters','pos',[50 50 900 600])
subplot(2,1,1);
plot(X1ss,YnoLineal,'.','Color','r');
hold on;
plot(X1,Y1,'Color','k');
title('Sistema + No linealidad');
ylabel('Y [V]')
xlabel('X1 [ug/m^3]')
legend('Y + Y orden 3','Y1 ideal')
axis([0 max(X1ss) 0 max(YnoLineal)]);
grid on;

% Al repetir el ajuste de curva con una recta se aprecia que los valores
% correspondientes a los errores generados por la App de Curve Fitting
% aumentaron, es decir, una regresion lineal no es optima para expresar el
% sistema en todo su dominio, se podria hacer uso de una regresion
% polinomial de mayor orden para obtener mejores resultados en el ajuste.

%% Item 1-E
% Calcular Estadisticas en el conjunto de experimentos

% Calcularemos las medias de conjuntos de 20 (Numero de experimentos)
% valores correspondientes a las imagenes de cada valor de X obteniendo
% un vector de tamaño M (Numero de datos por experimento), ya que tendra
% mas sentido estos resultados a la hora de hallar Errores Absolutos y
% Relativos

Ymeans = mean(Yexp);
Ystds = std(Yexp);
Yvars = var(Yexp);

% Dibujamos el vector Ymeans que contiene la media de cada conjunto de 20
```

```
% datos correspondiente a cada experimento realizado, esto nos permite
% apreciar que tanto se acerca el valor medio de las medidas a nuestro
% sistemas ideal. Pero no brinda informacion sobre que tan dispersas estan
% las medidas para cada valor de X. Describir el comportamiento de
% nuestro sistema basado en este grafico seria inapropiado.

figure('rend','painters','pos',[50 50 900 600])
plot(X1ss,Ymeans,'.','Color','r');
hold on;
plot(X1,Y1,'Color','k');
title('Ymeans vs X');
ylabel('Ymeans [V]')
xlabel('X1 [ug/m^3]')
legend('Ymeans','Y1 ideal')
axis([0 max(X1ss) 0 max(Ymeans)]);
grid on;

% Haciendo uso del vector de desviacion estandar podemos generar un grafico
% con barras de error para cada medida, de la siguiente manera:

figure('rend','painters','pos',[50 50 900 600])
errorbar(X1ss, Ymeans, Ystds,'.','Color','r')
title('Grafico Ymeans vs X con barras de error');
ylabel('Ymeans [V]')
xlabel('X1 [ug/m^3]')
axis([0 max(X1ss) 0 max(Ymeans)]);
grid on;

% En esta grafica podemos observar que las barras correspondientes a la
% desviacion estandar concuerdan con la dispercion de las muestras para
% diferentes valores de X1ss, es decir, debido al error introducido en la
% sensibilidad del sistema se aprecia que para valores mas altos de X, las
% medidas tomadas en los diferentes experimentos estan mas dispersas.
% Este grafico nos permite describir a nuestro sistema con mayor
% detalle.

%% Procedemos a calcular los errores absoluto y relativo de cada una de las
% medidas adquiridas en los experimentos.
for i=1:1:20
    EAbsMatrix(i,:) = abs(Yexp(i,:) - Ymeans);
    ERelMatrix(i,:) = (abs(Yexp(i,:) - Ymeans))./Ymeans;
end

% Para poder observar en un grafica el comportamiento del error del sistema
% generamos un vector de Errores Absolutos y uno de Errores Relativos,
% promediando los Errores de cada conjunto de 20 medidas (20 experimentos)
% para cada tipo de error por separado.

EAbsArray = mean(EAbsMatrix);
ERelArray = mean(ERelMatrix)*100;

% Graficamos los errores absoluto y relativo
figure('rend','painters','pos',[50 50 900 600])
plot(X1ss,EAbsArray,'.','Color','b');
title('Grafico de Errores Absolutos');
```

```
ylabel('EAbsArray [V]')
xlabel('X1 [ug/m^3]')
axis([0 max(X1ss) 0 max(EAbsArray)]);
grid on;

figure('rend','painters','pos',[50 50 900 600])
plot(X1ss,ERelArray,'.','Color','b');
title('Grafico de Errores Relativos');
ylabel('ERelArray [%]')
xlabel('X1 [ug/m^3]')
axis([0 max(X1ss) 0 max(ERelArray)]);
grid on;

% Se puede evidenciar que el error absoluto aumenta conforme aumenta X,
% pero no con una pendiente constante, es decir la velocidad con la que
% aumenta el error absoluto es mayor conforme aumenta X, esto se debe
% precisamente al error añadido a la sensibilidad del sistema.

%% Por ultimo calculamos el valor del error absoluto y relativo para todo el conjunto
de datos (20 experimentos)
% Lo cual nos permite ver de una manera mucho mas general el error de
% nuestro sistema.

disp('Error Absoluto [V]')
EAbs = mean(mean(EAbsMatrix))

disp('Error Relativo [%]')
Erel = mean(mean(ERelMatrix))*100
```