

Clase 2 de Taller: Aprendiendo Octave

¿Hicieron la presentación de deberes?

Pasen al pizarrón y expliquen su ejercicio

¿Qué es un SCRIPT?

Es una secuencia de instrucciones escritas en un lenguaje de programación.

Por ejemplo: uno que a partir de una fecha te diga qué día de la semana fue

```
% Solicitar la fecha al usuario
fecha_str = input('Ingrese la fecha en formato dd/mm/aaaa: ', 's');

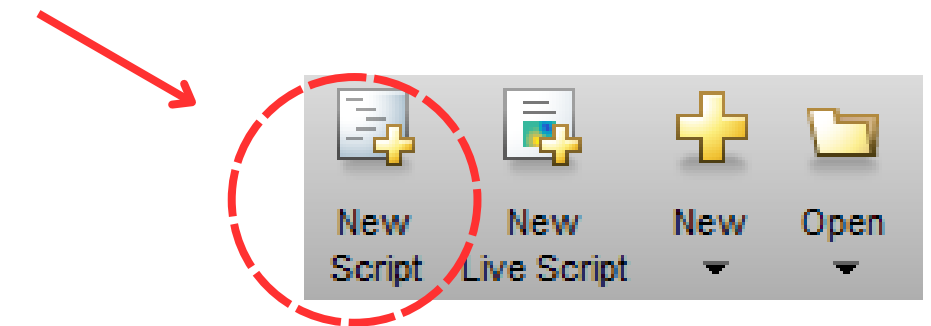
% Verificar si la entrada es válida
try
% Convertir la cadena de fecha a un valor numérico de fecha
fecha_num = datenum(fecha_str, 'dd/mm/yyyy');

% Obtener el día de la semana como un número (1 = Domingo, 2 = Lunes, ..., 7 = Sábado)
dia_semana_num = weekday(fecha_num);

% Convertir el número del día de la semana a un nombre
dias_semana = {'Domingo', 'Lunes', 'Martes', 'Miércoles', 'Jueves', 'Viernes', 'Sábado'};
dia_semana_nombre = dias_semana{dia_semana_num};

% Mostrar el resultado
fprintf('La fecha %s corresponde al día %s.\n', fecha_str, dia_semana_nombre);

catch
% Si hay un error al convertir la fecha, mostrar un mensaje de error
fprintf('Error: Formato de fecha incorrecto. Por favor, ingrese la fecha en formato dd/mm/aaaa.\n');
end
```



Ejercicios básicos 1:

Cree un script y nómbrelo '**calculos_sencillos.m**'. En él realice las siguientes operaciones

- Defina las variables $a=10$, $b=4$
- Calcule la suma, la diferencia, el producto y el cociente.
- Utilice `disp(['La suma a+b=', num2str(a+b)])` para imprimir el resultado en Command Windows de cada una de las operaciones
- Guarde el script. Borre todo con **clear** y luego ejecute el script desde Command Windows escribiendo '**calculos_sencillos**'

Recordatorio de cómo crear vectores

“Defina un vector de **tiempo** desde **0** a **10**, con un paso de **0.05**”

Opción 1:

```
t0 = 0
tf = 10
dt = 0.05

>> tiempo = [t0: dt: tf];
```

Opción 2: con **linspace()**

```
t0 = 0
tf = 10
dt = 0.05

num_elementos = round(1 + (tf-t0)/dt )

>> tiempo = linspace(t0, tf, num_elementos )
```

Ejercicios básicos 2:

Cree un script y nómbrelo 'guardar_datos.m'. En él realice las siguientes operaciones

- Defina un vector de **tiempo** desde **0** a **10**, con un paso de **0.05** y la variable **alfa=2.5**
- Calcule **Y=exp(alfa.*tiempo)** y guarde con el comando **save('t.mat', 'tiempo')** ambos vectores como matrices independientes.

Cree un script y nómbrelo 'cargar_datos.m'. En él realice las siguientes operaciones

- Borre el espacio de trabajo con **clear** y **clc**
- Utilice **load('t.mat')** para cargar el vector tiempo
- Utilice **disp(t(1))** y **disp(t(end))** para imprimir el tiempo inicial y final
- Con **disp** y **length** intente imprimir el largo del vector tiempo, es decir, el número de elementos.

Ejercicios básicos 3:

Cree un script y nómbrelo 'datos_en_txt.m'. En él realice las siguientes operaciones

- Defina dos variables **M=4** y **N=2**
- Con las funciones **zeros(M,N)**, **ones(M,N)**, **rand(M,N)**, cree distintas matrices
- Utilice **dlmwrite('random.txt','random')** para guardar las matrices en **.txt**
- Utilice **save()** para guardarlas como **.mat**

Color, Symbols, and Line Types

- Use "help plot" to find available Specifiers

Colors		Symbols		Line Types	
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star		
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Ejercicios básicos 4:

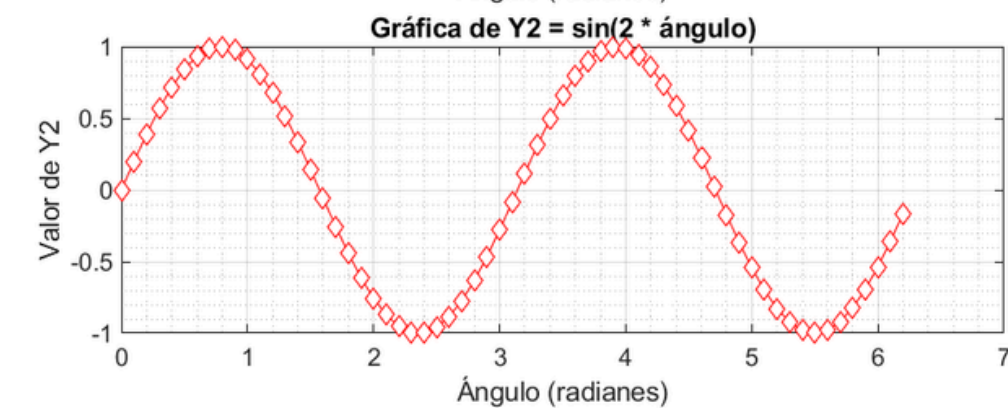
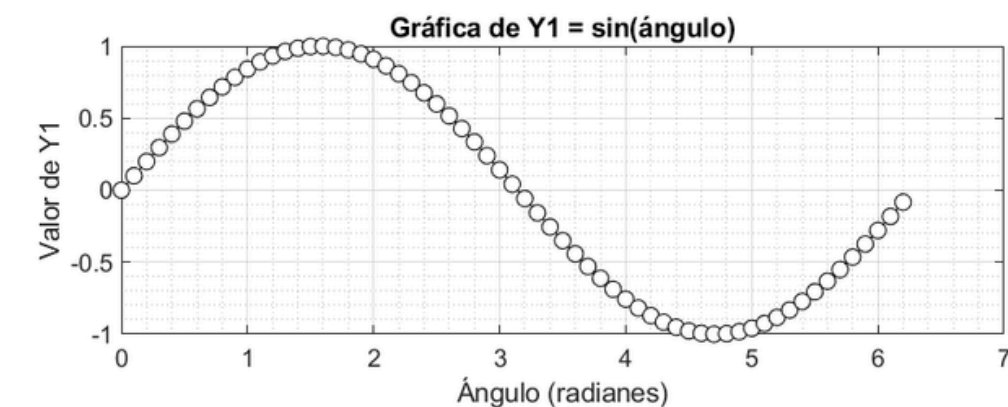
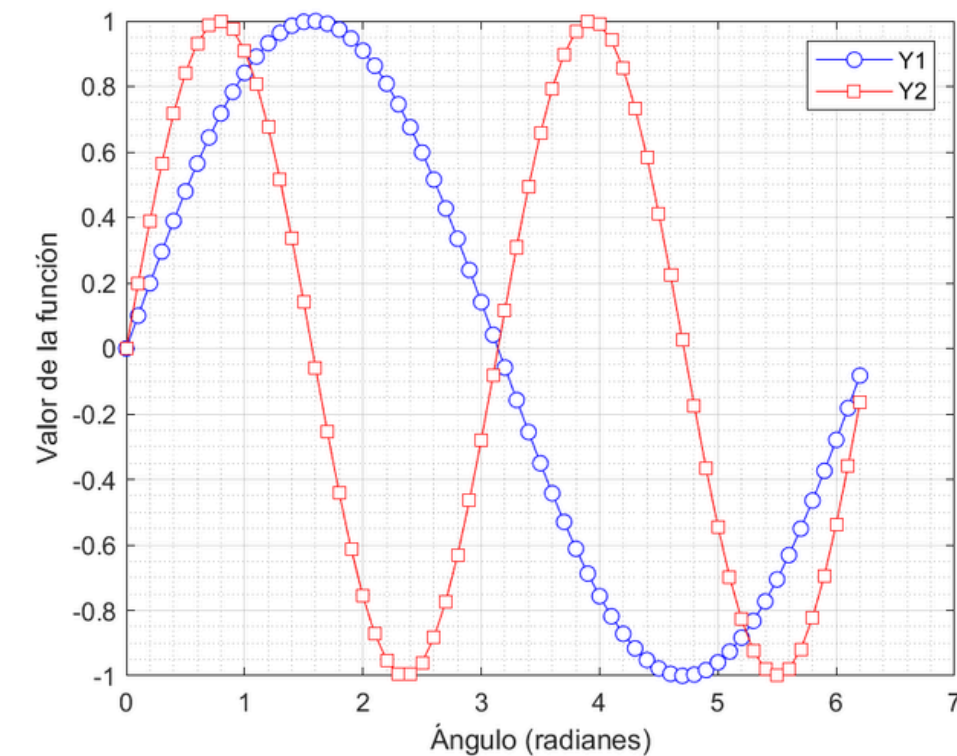
Cree un script y nómbrelo 'script_plots.m'. En él realice las siguientes operaciones

- Cree un vector de **angulos** de **0** a **2π** con incrementos de **0.1**.
- Defina **Y1= sin(angulos)**, **Y2=sin(2.*angulos)**
- En un mismo **figure()**, grafique **Y1** e **Y2** utilizando **plot** y **hold on**
- Investigue los comandos **xlabel** e **ylabel** para dar nombre a los ejes
- Utilice **grid on**, **grid minor**, **box on** para ver cómo se modifica la figura
- Utilice **legend('Y1','Y2')** para agregar una leyenda

- Investigue el comando **subplot** y grafique a **Y1** e **Y2** en subfiguras distintas.
- Guarde la figura en distintos formatos con **save** y **saveas**

Ayuda para la parte 1:

```
1 % script_plots.m
2 % Crear un vector de ángulos de 0 a 2*pi con incrementos de 0.1
3 angulos = 0:0.1:2*pi;
4
5 % Definir Y1 y Y2
6 Y1 = sin(angulos);
7 Y2 = sin(2 .* angulos);
8
9 % Graficar Y1 e Y2 en la misma figura
10 figure;
11 plot(angulos, Y1, 'b-o', 'DisplayName', 'Y1'); % 'b-' para línea azul
12 % 'o' para que los markers
13 % sean círculos
14 hold on; % Retener la gráfica actual
15 plot(angulos, Y2, 'r-s', 'DisplayName', 'Y2'); % 'r-' para línea roja
16 % 's' para que los markers
17 % sean círculos
18
19 % Etiquetar los ejes
20 xlabel('Ángulo (radianes)');
21 ylabel('Valor de la función');
22
23 % Ajustes visuales
24 grid on; % Activar la cuadrícula
25 grid minor; % Activar las líneas de cuadrícula menores
26 box on; % Mostrar el contorno de la caja
27
28 % Agregar leyenda
29 legend('Y1', 'Y2');
30
31 % Guardar la figura en distintos formatos
32 saveas(gcf, 'grafico_sin.png'); % Guardar como PNG
33 saveas(gcf, 'grafico_sin.fig'); % Guardar como figura de MATLAB
34
```



¿Cómo presentar figuras?

En el EVA tienen las pautas: [link](#)

Pautas para generar figuras

¿Qué es una figura?

- Un conjunto autocontenido con dos componentes:
- un gráfico: visual, cuantitativo (datos numéricos).
 - un texto corto que lo explica (pie de figura).

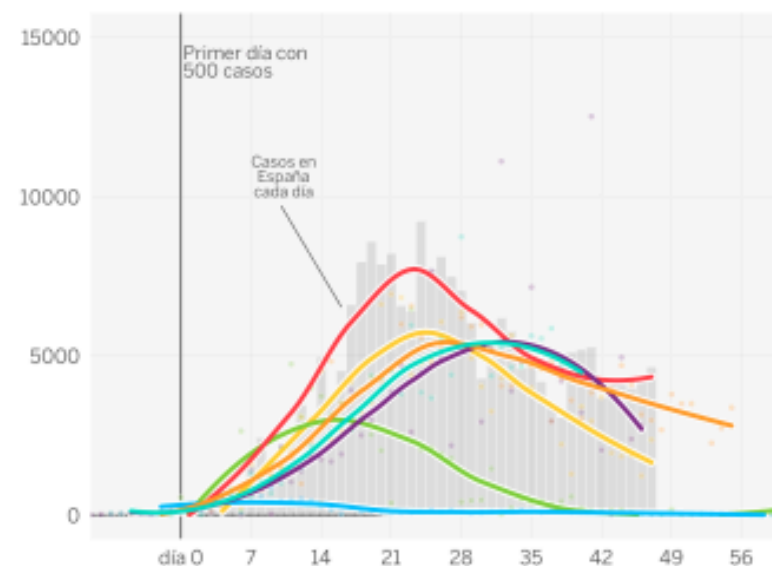
¿Por qué es importante una buena figura?

- Contiene información esencial del trabajo.
El cuerpo del texto del informe se centra en justificar, explicar y analizar las figuras.
Es uno de los ítems que se miran primero al leer un artículo.

¿Qué beneficios tiene?

- Resume y sintetiza mucha información.
Da mensajes claros, cualitativos y cuantitativos.
Se analiza rápidamente (tendencias, comparaciones, etc.)

Ejemplo de una figura incompleta:



¿Por qué está incompleta esta figura?

- No tiene nombre ni unidades en los ejes, por lo tanto no se sabe qué se está graficando.
No se sabe a qué corresponde cada línea de colores.
No tiene pie de figura.
¿Qué representan las barras grises?

Ejemplo de una figura completa

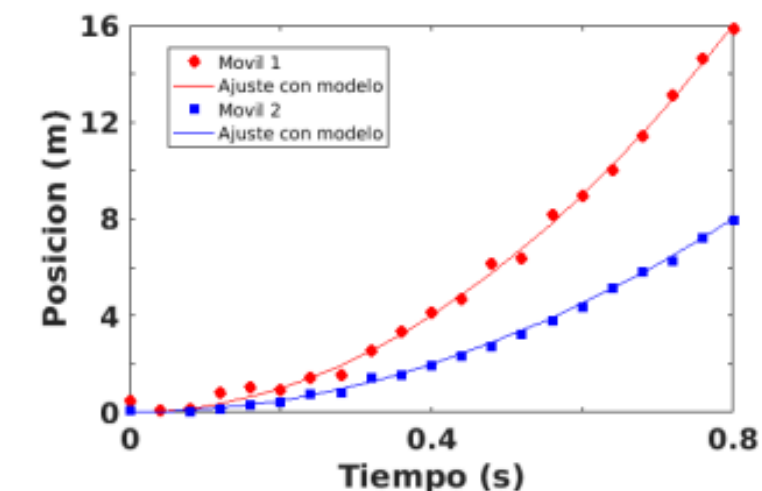


Figura 1. Posición en función del tiempo para dos móviles. Los símbolos rojos (azules) corresponden a la posición en función del tiempo para el móvil 1 (móvil 2). La línea continua roja (azul) corresponde al modelo teórico utilizado $x = \frac{1}{2}a^2$

Observen que la figura 1 contiene toda la información necesaria para interpretar los resultados obtenidos en las experiencias realizadas:

- Nombre y unidades en los ejes.
- Modelo utilizado para contrastar resultados.
- Pie de figura autocontenido.

Tarea para próxima clase:

Ahora que saben presentar Figuras, deben entregar las gráficas correspondientes al **1er experimento de caída libre**

- Gráfica de posición en función del tiempo
- Gráfica de velocidad en función del tiempo
- Intente utilizar **basic fitting** para determinar 'g' a partir de ambas gráficas

Recomendaciones:

- Pueden serle útiles los comandos **subplot**, **errorbar** (son datos experimentales), **legend**, **xlabel**, **ylabel**, **grid on**, **box on**, y utilizar la aplicación **basic fitting** para realizar un ajuste de los datos.

Fin de la Clase

Muchas gracias