

# Diplomado : Herramientas de Programación para Ciencias e Ingeniería

**Módulo: MATLAB (Clase 7)**

Docente: Juan Sebastián Salcedo Gallo

Universidad Nacional de Colombia Sede Manizales

# Contenido

- Análisis Básico de Datos (Estadística)
  - Consumo de Energía en Kew, Inglaterra.

# ¿Cómo leer un Dataset con MATLAB?

Para leer un conjunto de datos, usaremos **dataset**. Cuya sintaxis es:

```
A = dataset('file', nombre_del_archivo, 'ParamName', 'Value')
```

Este comando crea un arreglo de datos (A) con el par *Name/Value* especificado.

## **Pares *Name/Value***

Delimiter	
	A character vector or string scalar indicating the character separating columns in the file. Values are <ul style="list-style-type: none"><li>'\t' (tab, the default when no format is specified)</li><li>' ' (space, the default when a format is specified)</li><li>',' (comma)</li><li> ';' (semicolon)</li><li>' ' (bar)</li></ul>

# ¿Con qué vamos a trabajar?

Para eso, vayan a mi página de GitHub, busquen el repositorio ***MATLAB\_module*** y descarguen el archivo ***energy-consumption.csv***. Este conjunto de datos corresponde al consumo energético de una ciudad en Inglaterra (Kew) entre 2010 y 2018, en unidades de kWh. Las columnas representan la hora medida cada 30 minutos.

# ¿Qué vamos a hacer?

- Graficar el consumo energético en kWh para todos los días del primer mes del Dataset. (consumo vs. Hora para todo el primer mes, para todos los días del mes).
- Calcularemos la media de consumo para el primer mes, en función de la hora. (`media(consumo(1er_mes))`)
- Calcularemos la desviación estándar del consumo para el primer mes, en función de la hora (`std(consumo(1er_mes))`)
- Graficar la media con su respectiva barra de error(desviación estándar por hora) para el primer mes
- Ajustar un polinomio a la media, usando ***polyfit***
- Graficar la media, con su respectiva desviación estándar y el polinomio ajustado para el primer mes

- Graficar el consumo energético en kWh para todos los días del primer mes del Dataset. (consumo vs. Hora para todo el primer mes).

```
1 ds = dataset('file', 'energy-consumption.csv', 'delimiter', ',');
2
3 consumo = ds(:, 5:end-1);
4 figure(1)
5 hold on
6 hours = 0: 0.5:23.5;
7 labels = {};
8
9 for i = 1:31
10     plot(hours, consumo(i, :))
11     labels{i} = ['Day ', num2str(i)];
12 end
13
14 xlabel('Hours')
15 ylabel('Consumo [kWh]')
16 legend(labels,'Location', 'bestoutside')
17 grid
```

- Calcularemos la media de consumo para el primer mes, en función de la hora. (media(consumo(1er\_mes)))

Para eso usaremos el comando ***mean(A)***. Este comando retorna la media de la matriz A en la dimensión de interés.

Si A es un array multidimensional, ***mean(A)*** opera a lo largo de la primera dimensión del array cuyo tamaño no sea igual a 1, tratando los elementos como vectores. Esta dimensión se convierte en 1, mientras que los tamaños de todas las demás dimensiones permanecen iguales.

```
mean_consumption_january = mean(double(consumo(1:31, :)));
```

- Calcularemos la desviación estándar del consumo para el primer mes, en función de la hora (std(consumo(1er\_mes)))

Para eso usaremos el comando **std(A)**. Este comando retorna la desviación de los datos de la matriz A en la dimensión de interés.

Si A es una matriz multidimensional, **std(A)** opera a lo largo de la primera cota de matriz cuyo tamaño no es igual a 1, tratando los elementos como vectores. El tamaño de esta cota se convierte en 1 mientras que los tamaños de todas las demás cotas permanecen iguales.

```
standard_deviation_january = std(double(consumo(1:31, :)));
```



- Graficar la media con su respectiva barra de error(desviación estándar) para el primer mes

Para esto, usaremos el comando ***errorbar(x,y,err)***

***errorbar(x,y,err)*** grafica y versus x y dibuja una barra de error vertical en cada punto de datos.

```
figure(2)
subplot(1,2,1)
errorbar(hours, mean_consumption_january, standard_deviation_january, '-sr', 'MarkerSize', 7);
```

- Ajustar un polinomio a la media, usando ***polyfit***

Para esto usaremos el comando `polyfit`, visto en la clase anterior, podemos generar una variable adicional para dar cuenta del grado del polinomio y probar varios parámetros hasta ver cuál se ajusta mejor.

Para evaluar el polinomio ajustado, usamos el método `polyfit`, visto también en la clase anterior.

```
parameters_january = polyfit(hours, mean_consumption_january, order_polinomial);  
yfit_january = polyval(parameters_january, hours);
```

- Graficar la media, con su respectiva desviación estándar y el polinomio ajustado para el primer mes

```
figure(2)
subplot(1,2,1)
errorbar(hours, mean_consumption_january, standard_deviation_january, '-sr', 'MarkerSize', 7);
hold on
plot(hours, yfit_january, 'r--')
legend('January')
xlim([0 24])
ylim([200 1200])
xticks(0:24)
grid on
box on
ylabel('Power Consumption [kWh]')
xlabel('Time [h]')
```

# Ejercicio

Realice lo mismo, esta vez para un mes diferente (puede ser el mes siguiente), y compárelo con los resultados obtenidos para el mes de agosto del 2010.