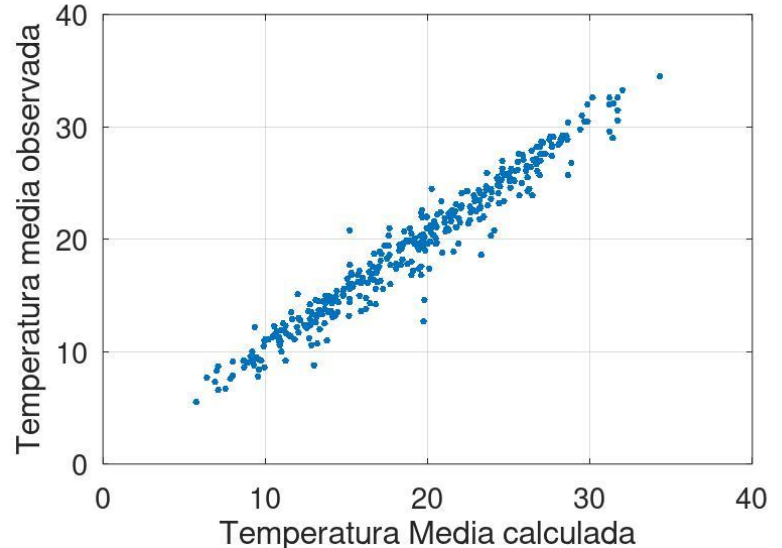


PLOTS Y ALGUNAS FUNCIONES AVANZADAS

Diagrama de dispersión

```
figure(1)
plot(TmedCal,TmedObs,'.','markersize',10);
ylabel ('Temperatura Media Observada °C');
xlabel ('Temperatura media Calculada °C');
grid on;
axis([0 40 0 40]);
title({'Diagrama de dispersión para la
Temperatura media';'Salto Grande 2022'});
set(gca,'FontSize',14)
```

Diagrama de dispersión para la Temperatur media
Salto Grande 2022



Estructura *for*

La estructura *for*, permite repetir instrucciones una cantidad N de veces.

Si se especifica `i = 1:10` la estructura se repetirá 10 veces.

```
for i = 1:10
    disp("hola")
endfor

for i = 1:10
    disp(i)
endfor

# Análogamente
valores = 1:10
for i = valores
    disp(i)
endfor

# Si se utiliza una matriz
# ¿Cuál puede ser el resultado?

matriz = [1 2 3 4 5; 6 7 8 9 10; 11 12 13 14 15]
for i = valores
    disp(i)
endfor

# Probar estos ejemplos
```

Redimensionamiento

Supongamos ahora que tenemos un vector con datos **horarios** a lo largo de un año para una estación.

Generalmente puede ser muy útil distribuir los datos del vector unidimensional a una mayor dimensión.

Por ejemplo, **pasar de un vector a una matriz**



Cada fila y/o columna pueden tener algún sentido interesante para trabajar.

Redimensionamiento

Si tenemos un vector que almacena la temperatura media horaria para alguna estación, en un año dado.

| | | | |
|-------|--------|--------|--|
| T_med | double | 8760x1 | [19.180; 19.180; 18.340; 16.910; 17.150; 16.740; 19.080; ... |
|-------|--------|--------|--|



¿En qué tipo de matriz puede ser productivo almacenar estos datos?

Redimensionamiento

Un posible reordenamiento puede ser agrupar en filas todos los datos de un día y en columnas cada día del año.

```
M_Tmed = reshape(T_med, [24, 365]);
```

filas

columnas

La función `reshape()` recibe como parámetros la variable de datos y la nueva “forma” como un vector. En este caso de 24 filas y 365 columnas.

M_Tmed

double

24x365

[18.270, 17.210, ...]



Operación de funciones sobre matrices

Con los datos ordenados de esta forma, es posible calcular de forma muy fácil (por ejemplo) las medias diarias o medias horarias a lo largo de todo el año.



```
M_Tmed = reshape(T_med, [24, 365]);  
Media_diaria = mean(M_Tmed);  
Media_horaria = mean(M_Tmed, 2);
```

| | | | |
|---------------|--------|-------|--|
| Media_diaria | double | 1x365 | [21.063, 21.657, 23.878, 28.432, 22.983, 20.561, 21.620, ... |
| Media_horaria | double | 24x1 | [14.424; 14.094; 13.776; 13.506; 13.263; 13.070; 13.310; ... |

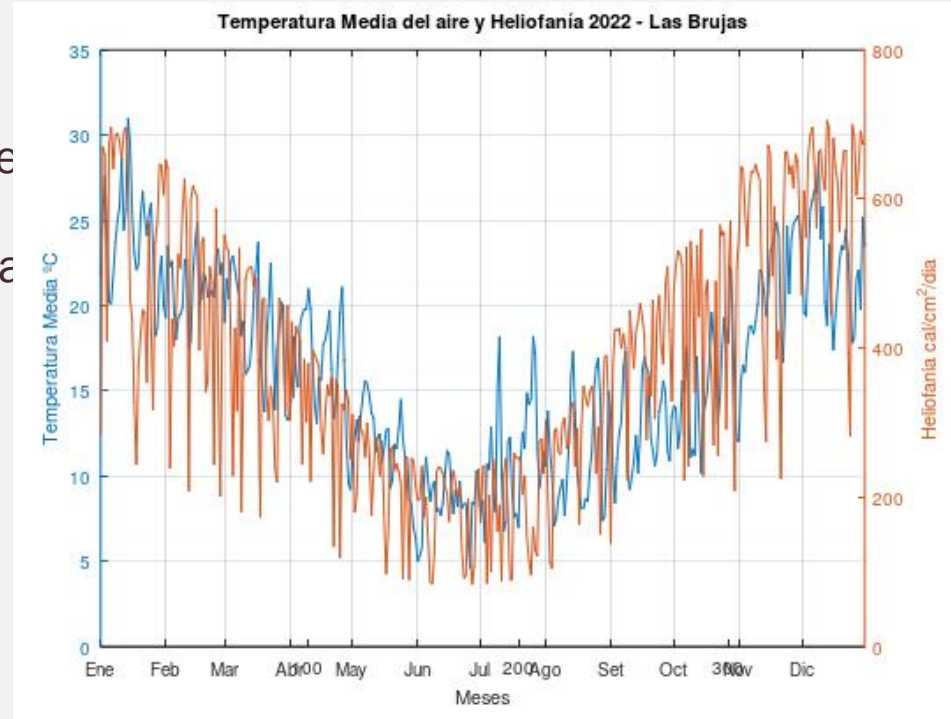


Ploteo de dos magnitudes diferentes en una misma gráfica

Supongamos que tenemos datos de temperatura media y heliofanía en una misma gráfica, para poder respetar que la magnitud es diferente necesitamos utilizar dos ejes diferentes para expresar las magnitudes.

Para esto utilizamos la función:

```
plotyy()
```



Ploteo de dos magnitudes diferentes en una misma gráfica

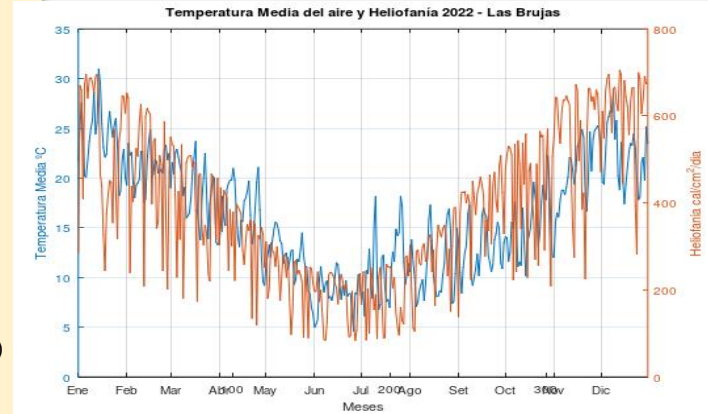
```
figure(1)
dias = 1:365;
[ax,hline1,hline2] = plotyy(dias,D_Tmedia(:,4),dias,D_helio(:,4));

ylabel(ax(1), 'Temperatura Media °C');
ylabel(ax(2), 'Heliofanía cal/cm^2/día');

set(hline1, 'linewidth',2)
set(hline2, 'linewidth',2)
set(ax(1), 'linewidth',2);
set(ax(2), 'linewidth',2);

grid on
title('Temperatura Media del aire y Heliofanía 2022 - Las Brujas')
xlabel('Meses')

xlim([1,365])
iniMes = find(D_Tmedia(:,1)==1);
xticks(iniMes)
NMeses = {"Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago",
"Set", "Oct", "Nov", "Dic"}
xticklabels(NMeses);
set(haxes, 'FontSize',14);
```



Mediante las variables [ax,hline1,hline2] se ajustan propiedades de la gráfica.

En las siguientes slides se pueden ver ejemplos de código completos.

Cada ejemplo utiliza una forma diferente de “dibujar” las etiquetas de tiempo en las figuras.

```

1  clc # Borra a ventana de comandos
2  clear all # Borra todas las variables en memoria
3  close all # Cierra todas las ventanas abiertas
4
5  # Cargo los datos y los ordeno
6
7  Datos = load("Datos_INIA_procesados.txt");
8  Datos = flip(Datos);
9
10 # Se pretenden visualiazar en una misma diapositiva la heliofanía y la temperatura
11 # de un año.
12 # Para este ejemplo se cargan los datos como lo realizado en la entrega 1
13 % Heliofanía
14 D_helio = Datos(:, [1 2 3 4]);
15 % Temperatura Media
16 D_Tmedia = Datos(:, [1 2 3 5]);
17
18 # Por inspección se puede ver que es un año de 365 días
19 dias = 1:365;
20 # Genero la figura
21 figure(1)
22 ax = plotyy(dias, D_Tmedia(:,4), dias, D_helio(:,4));
23 ylabel (ax(1), "Temperatura Media °C");
24 ylabel (ax(2), "Heliofanía cal/cm^2/dia");
25 grid on
26 title('Temperatura Media del aire y Heliofanía 2022 - Las Brujas')
27 xlabel('Meses')
28 xlim([1,365]);
29 iniMes = find(D_Tmedia(:,1)==1);
30 xticks(ax(1), iniMes); # Se asigan los valores del dominio a mostrar en el eje
31 xticks(ax(2)); # Para no superponer las etiquetas, no se determinan los valores para ax(2)
32 NMeses = {"Ene", "Feb", "Mar", "Abr", "May", "Jun", "Jul", "Ago", "Set", "Oct", "Nov", "Dic"};
33 xticklabels(NMeses); # Se asigan las etiquetas para cada valor indicado en iniMes

```

```
35 DATOS = load("Brujas_hora.txt");
36
37 # Temperatura Media para cada hora del año
38 T_med = DATOS(:,6);
39
40 # Para un año, mes y día genérico, genero un vector horario
41 # solo importa la hora para este ejemplo
42 hora = datenum(1,1,1,0:23);
43
44 # Se genera una matriz donde cada fila determina una hora,
45 # y cada columna un día del año.
46 M_Tmed = reshape(T_med, [24,365]);
47
48 # Se calcula la media sobre la dimensión 2 para obtener
49 # la media de temperaturas para cada hora del día.
50 Media_horaria = mean(M_Tmed,2);
51
52 figure(2)
53 grid on
54 plot(hora,Media_horaria);
55 grid on
56 title('Temperatura Media para el año 2022')
57 xlabel('Hora')
58 ylabel('Temperatura °C')
59 datetick('x','HH:MM','keeplimits')
```