

Laboratório de Controle - Aula 2 - 2022/1

Introdução ao Matlab

Nome:

```
turma=3;  
I=1;  
datetime('now')
```

```
ans = datetime  
19-May-2022 18:46:17
```

```
pwd
```

```
ans =  
'C:\Users\diona\OneDrive\Área de Trabalho\ufes\Laboratorio de Controle Automático\Aula2'
```

Link para documento sobre o [Matlab](#)

Importante: nos comandos abaixo, não use ponto e vírgula após comandos que geram resultados que devem aparecer no relatório!

Afinal, é o resultado que mostra se o comando está correto!

Atividade 1: Comandos para manipular variáveis

1.1 Criar um número complexo $z = x + jy$, com $x \neq 0$, $y \neq 0$ e calcular seu valor absoluto.

```
z = 3 + 4j
```

```
z = 3.0000 + 4.0000i
```

```
abs(z)
```

```
ans = 5
```

1.2 Definir uma matriz A qualquer com dimensão 4x4 e com determinante diferente de zero (Dois comandos, e mostrar a matriz e seu determinante).

```
A=[1 3 5 9;1 3 1 7;4 3 9 7;5 2 0 9]
```

```
A = 4x4  
    1     3     5     9  
    1     3     1     7  
    4     3     9     7  
    5     2     0     9
```

```
det(A)
```

```
ans = -376
```

1.3 Definir um vetor b não nulo com dimensão 4×1 e então obter x que satisfaz $A \cdot x = b$ (dois comandos)

```
b = [2 2 2 2]
```

```
b = 1×4  
    2    2    2    2
```

```
x = b/A
```

```
x = 1×4  
-1.0106    1.1170    0.6596   -0.1489
```

1.4 Salvar apenas as variáveis A, x, b em um arquivo (seu nome).mat

```
save Dionatas.mat A x b
```

1.5 Calcular o polinômio característico de A e depois suas raízes (dois comandos e mostrar)

```
p=poly(A)
```

```
p = 1×5  
    1.0000   -22.0000    71.0000   287.0000  -376.0000
```

```
roots(p)
```

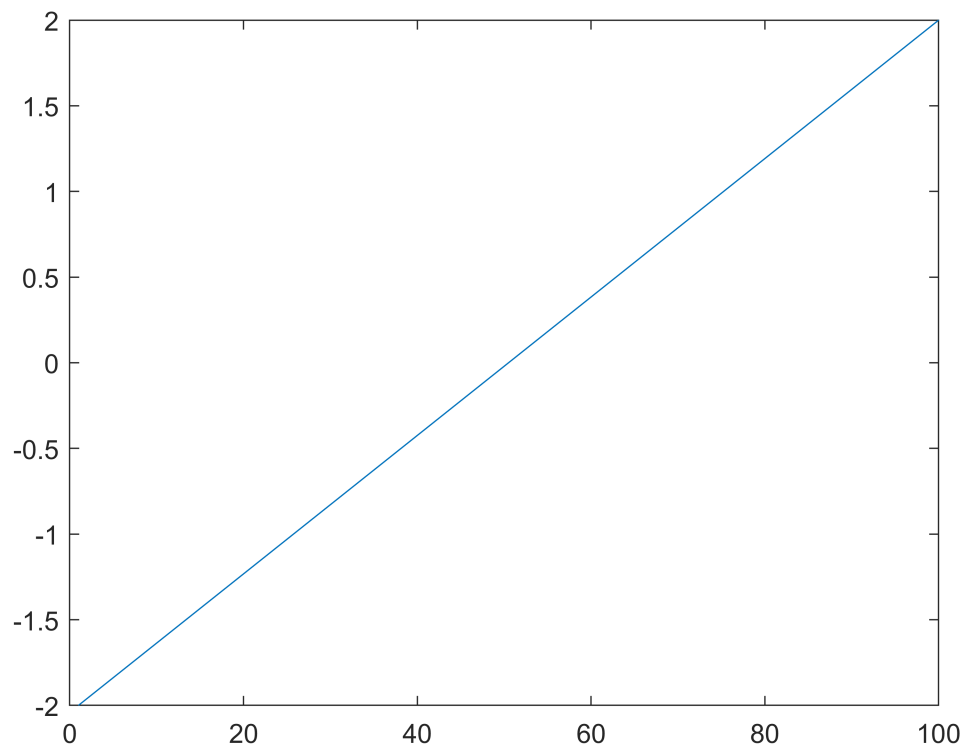
```
ans = 4×1  
    16.8564  
     6.9427  
    -2.9050  
     1.1060
```

1.6 Gerar um vetor z com 100 elementos uniformemente espaçados de -2 a 2. Usar ; após o comando, e mostrar z com o comando `plot(z)`.

```
z=linspace(-2,2,100)
```

```
z = 1×100  
-2.0000   -1.9596   -1.9192   -1.8788   -1.8384   -1.7980   -1.7576   -1.7172 ...
```

```
plot(z)
```



1.7 Criar uma variável M do tipo struct com os campos (fields) nome, turma, I, data (dia:mês:ano) e atribuir valores aos mesmos

```
M = struct('nome',[], 'turma',[], 'I',[], 'data',[])
```

```
M = struct with fields:
  nome: []
  turma: []
  I: []
  data: []
```

```
M.nome= 'Dionatas';
M.turma= 3;
M.I= 1;
M.data='11:01:2000';
M
```

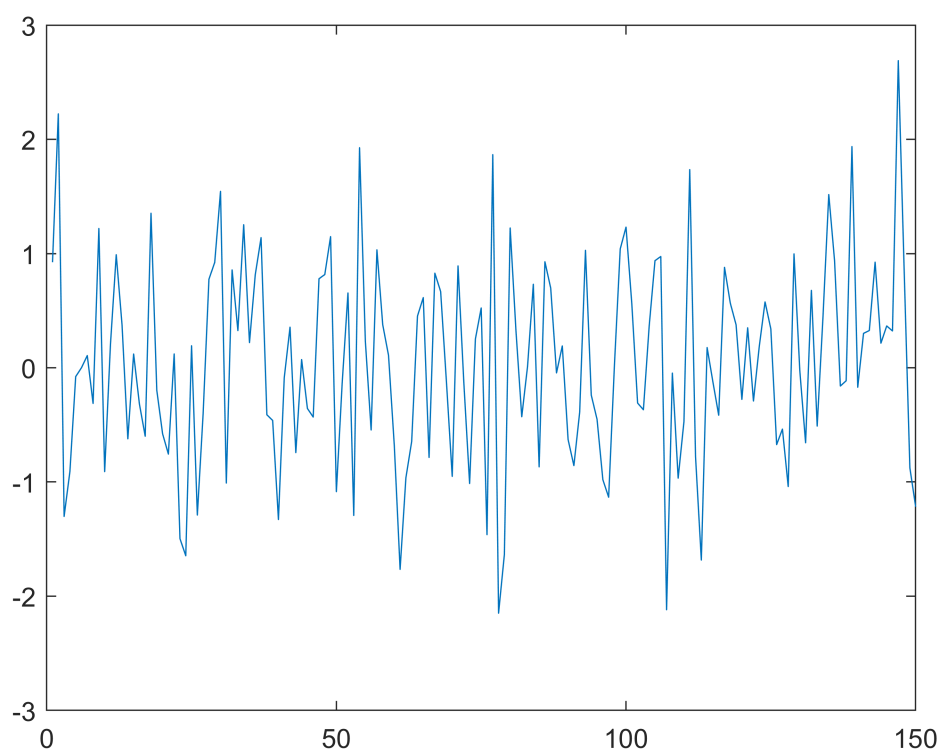
```
M = struct with fields:
  nome: 'Dionatas'
  turma: 3
  I: 1
  data: '11:01:2000'
```

1.8 Criar um vetor y com 150 elementos vindos de uma distribuição normal. Usar ; após o comando. Plotar então o histograma de y para confirmar sua distribuição normal.

```
y=randn(150,1)
```

```
y = 150x1  
0.9239  
2.2231  
-1.3019  
-0.9090  
-0.0775  
0.0005  
0.1069  
-0.3125  
1.2196  
-0.9096  
:  
:
```

```
plot(y)
```



1.9 Usar um comando apenas **para contar quantos** valores de y são maiores que zero.

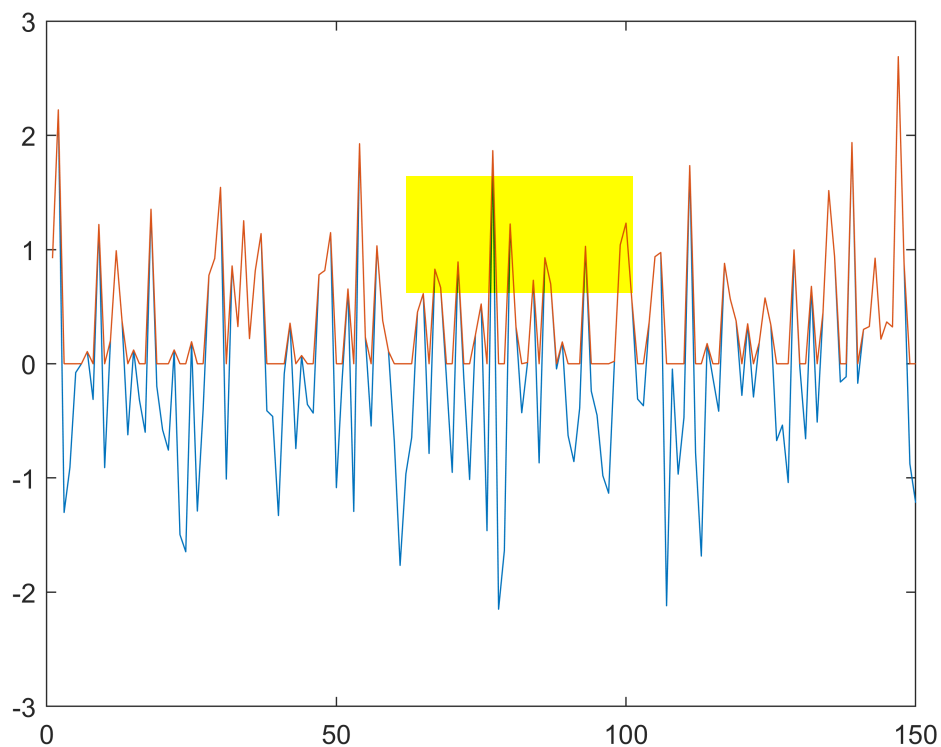
```
find(y>0)
```

```
ans = 78x1  
1  
2  
6  
7  
9  
11
```

12
13
15
18
:
:

1.10 Com no máximo dois comandos, substituir todos elementos negativos de y por zero, gerando $y1$. Plote `plot([y y1])` para confirmar o sucesso dos comandos.

```
y1=y;  
y1(y1<0)=0;  
plot([y y1])
```



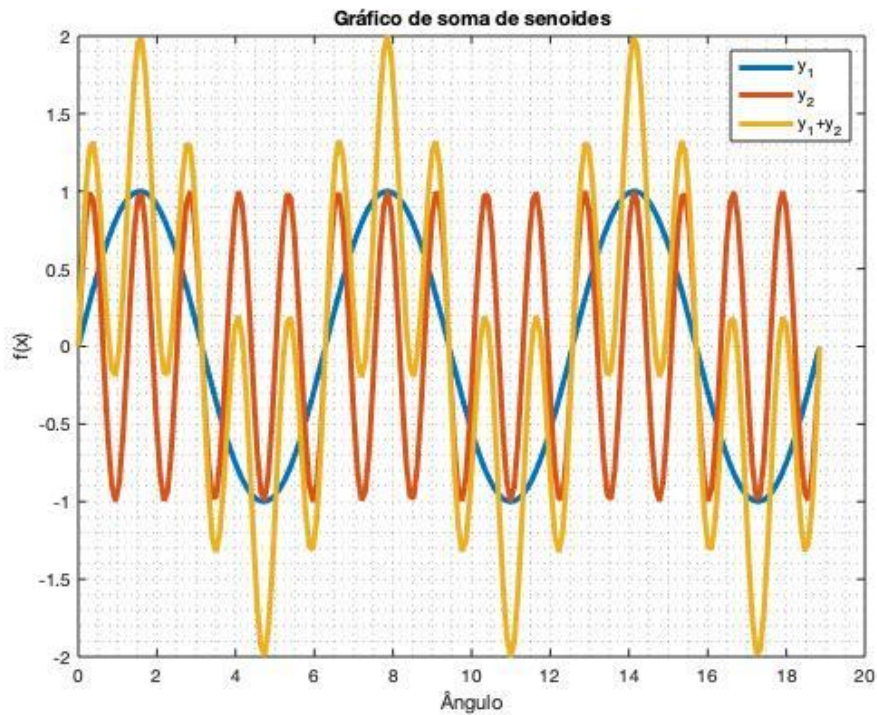
Atividade 2: Comandos para plotar dados e respostas de sistemas

Importante: Ao usar comandos para gerar figuras no relatório, comece com o comando `figure`, que abre uma nova figura, garantindo assim que nada será plotado sobre figuras já existentes.

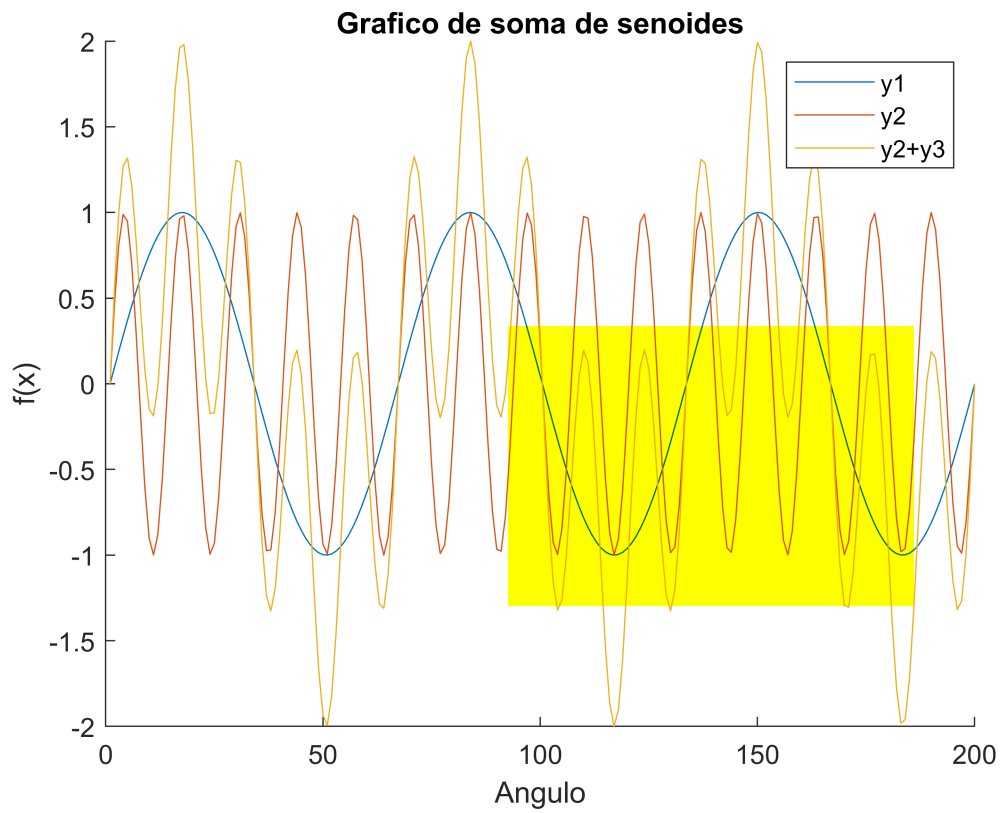
Sejam as variáveis $x, y1, y2, y3$.

```
x=linspace(0,6*pi,200); y1=sin(x); y2=sin(5*x); y3=y1+y2;
```

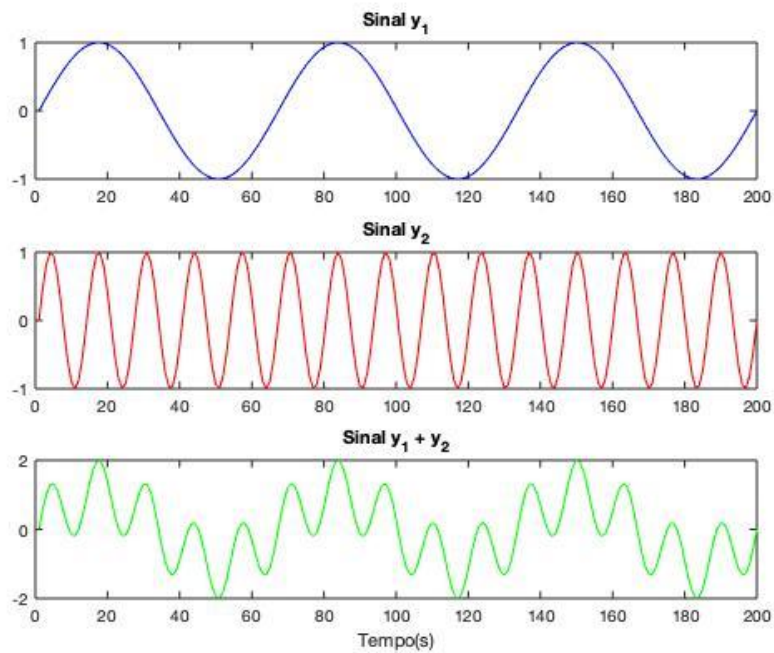
2.1 Dê os comandos para plotar y_1, y_2, y_1+y_2 exatamente como na figura mostrada. Ou seja, gere uma figura igual!



```
figure
hold on
title('Grafico de soma de senoides');
plot(y1)
plot(y2)
plot(y3)
legend('y1', 'y2', 'y2+y3');
ylabel('f(x)');
xlabel('Angulo');
```



2.2 Dê os comandos para produzir uma figura como a mostrada abaixo (Dica: subplot)

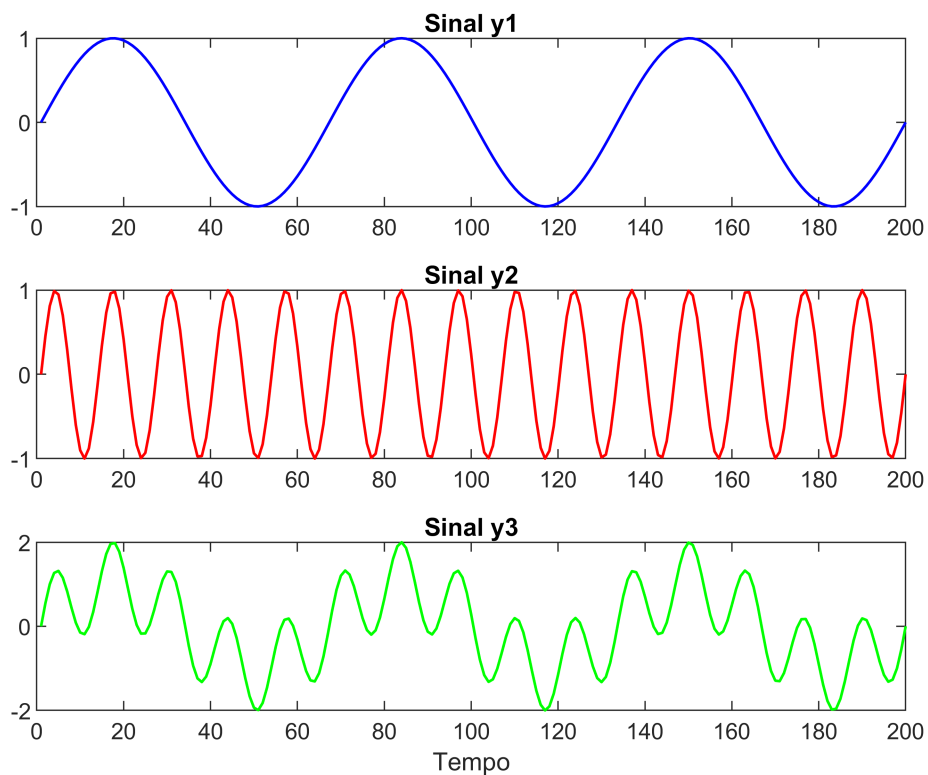


figure

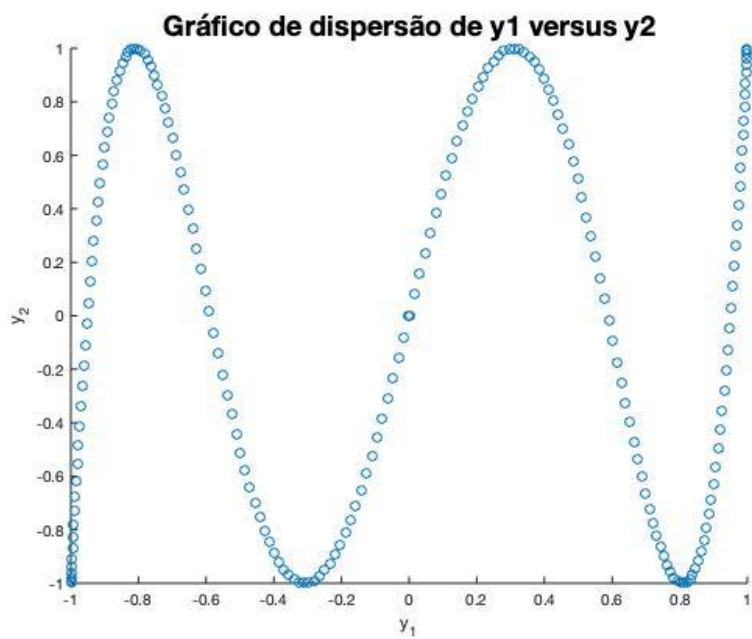
```
subplot(3,1,1)
plot(y1,'b','LineWidth',1);
title('Sinal y1');
```

```
subplot(3,1,2)
plot(y2,'r','LineWidth',1);
title('Sinal y2');
```

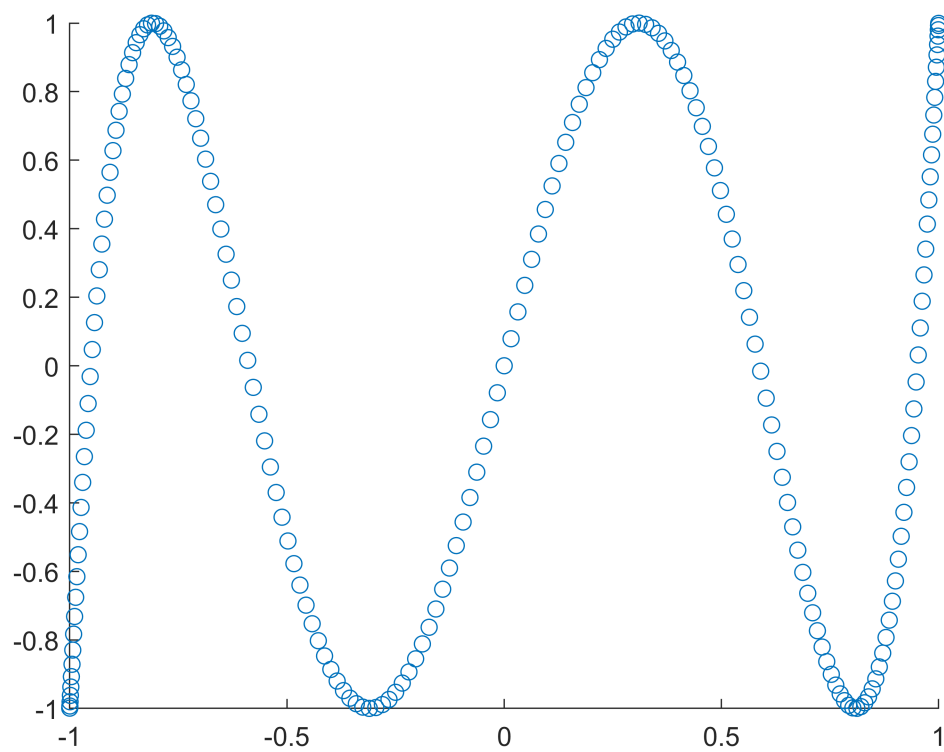
```
subplot(3,1,3)
plot(y3,'g','LineWidth',1);
title('Sinal y3');
xlabel('Tempo');
```



2.3 Dê os comandos para gerar o gráfico de dispersão de y1 versus y2 conforme a figura mostrada (dica: scatter)



```
figure  
scatter(y1,y2)
```



2.4 Crie uma nova figura e plote y_1 e plote em todos os picos deste sinal o caractere 'X', em vermelho. As coordenadas dos picos devem ser obtidas na janela Command Window do Matlab antes de executar este código. Dica: ver `ginput`.

```
figure
```

2.5 Plote a figura abaixo (beijaflor.png) (Dica: `imshow`)

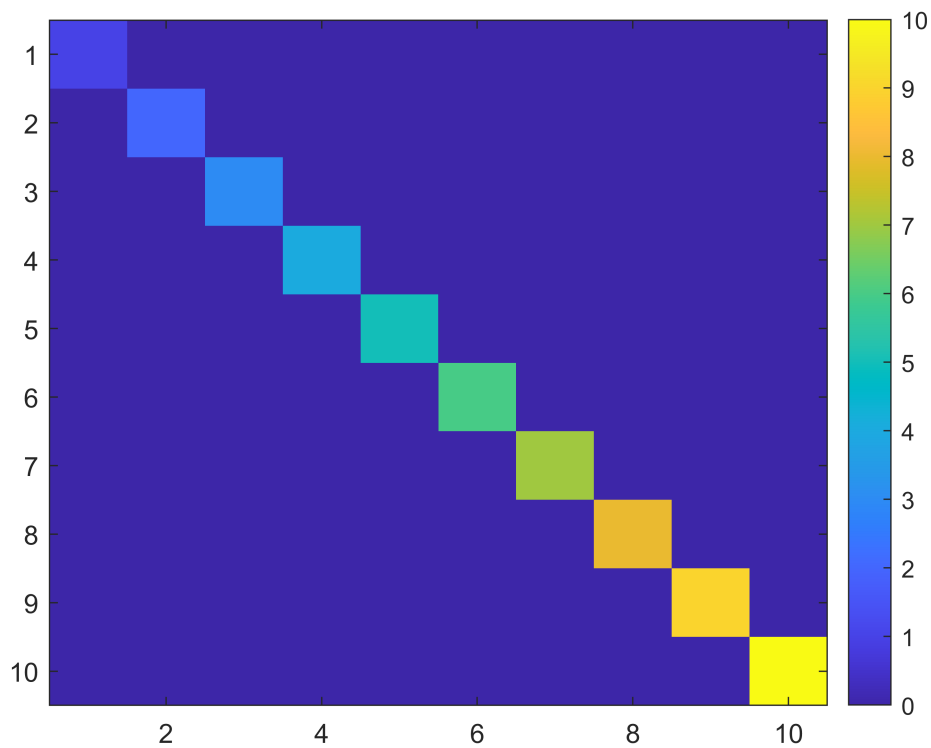


```
figure  
imshow('beijaflor.png')
```



2.6 Execute os comandos abaixo e descreva o que cada um deles faz

```
figure;  
C=diag([1:10],0);  
imagesc(C);  
colorbar;
```



Resposta:

C=diag([1:10],0)

Retorna uma matriz diagonal quadrada com os elementos do vetor v na diagonal principal, nesse exemplo é uma matriz 10x10.

```
C=diag([1:10],0)
```

C = 10x10

```

1   0   0   0   0   0   0   0   0   0
0   2   0   0   0   0   0   0   0   0
0   0   3   0   0   0   0   0   0   0
0   0   0   4   0   0   0   0   0   0
0   0   0   0   5   0   0   0   0   0
0   0   0   0   0   6   0   0   0   0
0   0   0   0   0   0   7   0   0   0
0   0   0   0   0   0   0   8   0   0
0   0   0   0   0   0   0   0   9   0
0   0   0   0   0   0   0   0   0  10

```

imagesc(C)

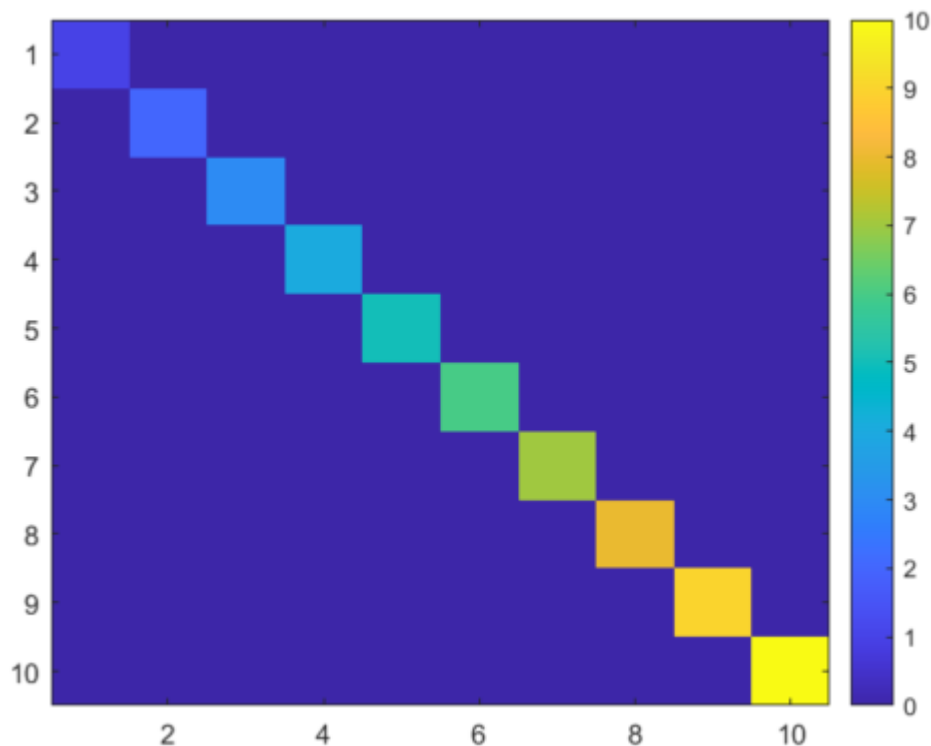
Exibe os dados de uma matriz (no caso do exemplo 10x10) como uma imagem que usa as cores referente ao mapa de cores do matlab.

Cada elemento de C especifica a core de um pixel da imagem.

colorbar:

O comando é referente a uma barra de cores vertical à direita dos eixos (ou do gráfico atual), essas barras exibem o mapa de cores atual e indicam o mapeamento dos valores de dados no mapa de cores

Nese exemplo:



A matriz possui diferentes elementos na sua diagonal (comando diag), cada cor se refere a um elemento (comando imagesc) e uma barra de cores (comando colobar).

2.7 Analisar o heatmap gerado pelo código abaixo, e comentar o que é mostrado.

Resposta:

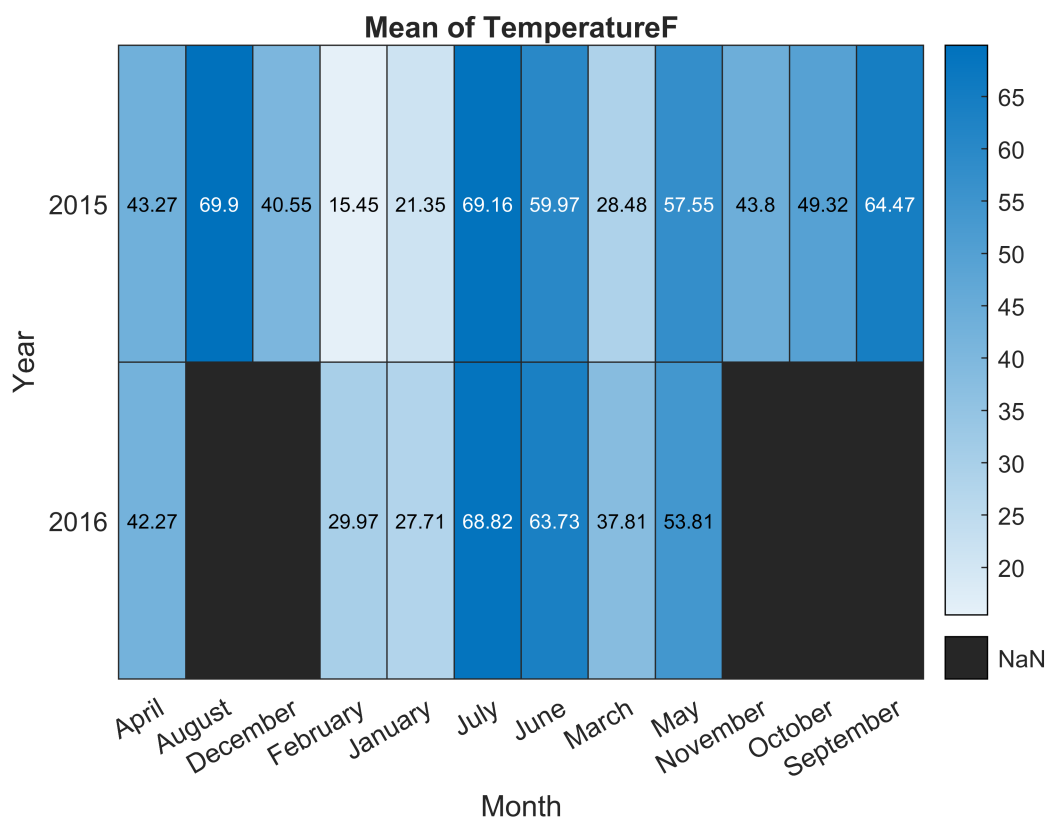
O `tbl` é uma matriz 5x4, contendo os dados do ano de 2015 referente a temperatura em "Fahrenheit".

```
%tbl = readtable(fullfile(matlabroot,'examples','graphics','TemperatureData.csv'));  
load dados_tbl.mat  
head(tbl,5)
```

ans = 5x4 table

	Year	Month	Day	TemperatureF
1	2015	'January'	1	23
2	2015	'January'	2	31
3	2015	'January'	3	25
4	2015	'January'	4	39
5	2015	'January'	5	29

```
h = heatmap(tbl,'Month','Year','ColorVariable','TemperatureF');
```



Resposta:

Nesse heatmap gerado, é mostrado as variações da temperatura nos anos de 2015 e 2016.

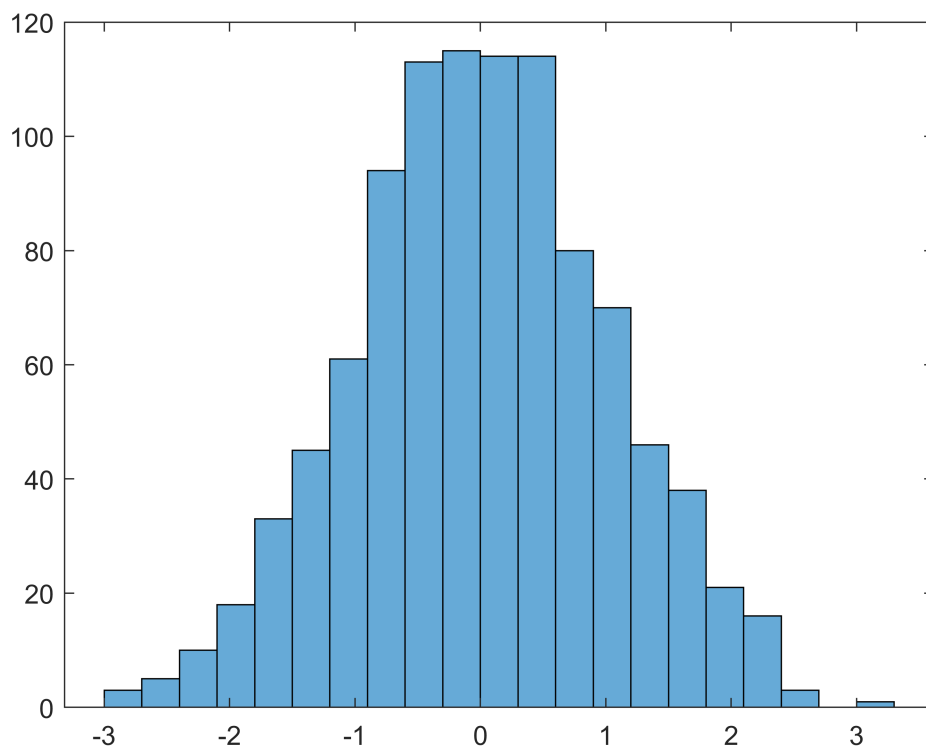
Foram obtidos amostras de cada mês e a cor varia em função da temperatura, quanto maior é o valor em "Fahrenheit" da temperatura, mais escuro será o tom do azul. Nos casos onde ocorre a cor preta, significa que não foi coletado dado. (NaN)

2.8 Criar um vetor w com 1000 valores aleatórios vindos de uma distribuição normal e plotar seu histograma com o comando `histogram`. Observe o **histograma e obtenha aproximadamente sua média**

```
w=randn(1000,1)
```

```
w = 1000x1
-1.6148
 1.8401
-0.8189
 0.9921
 0.5338
-1.5267
 2.0229
 0.5523
 1.8201
 0.3426
      ⋮
```

```
histogram(w)
```



```
mean(w)
```

```
ans = 0.0151
```

Resposta:

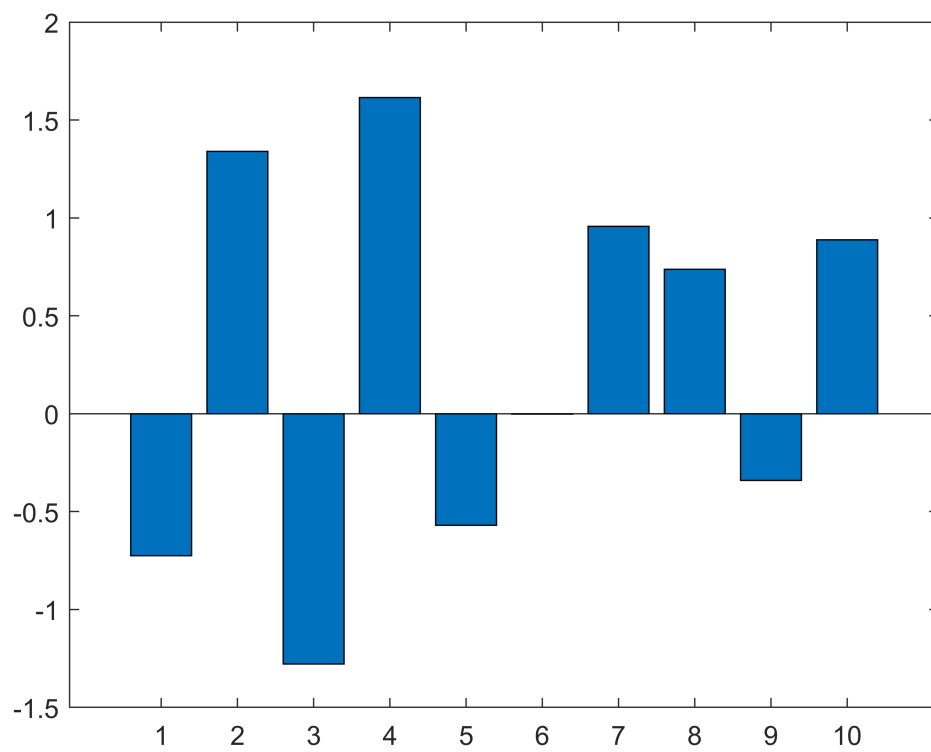
A média é obtida pelo comando "mean(w)".

2.9 Criar um vetor y com 10 valores aleatórios vindos de uma distribuição normal (randn) e plotar seus valores com o comando bar e depois com o comando stem

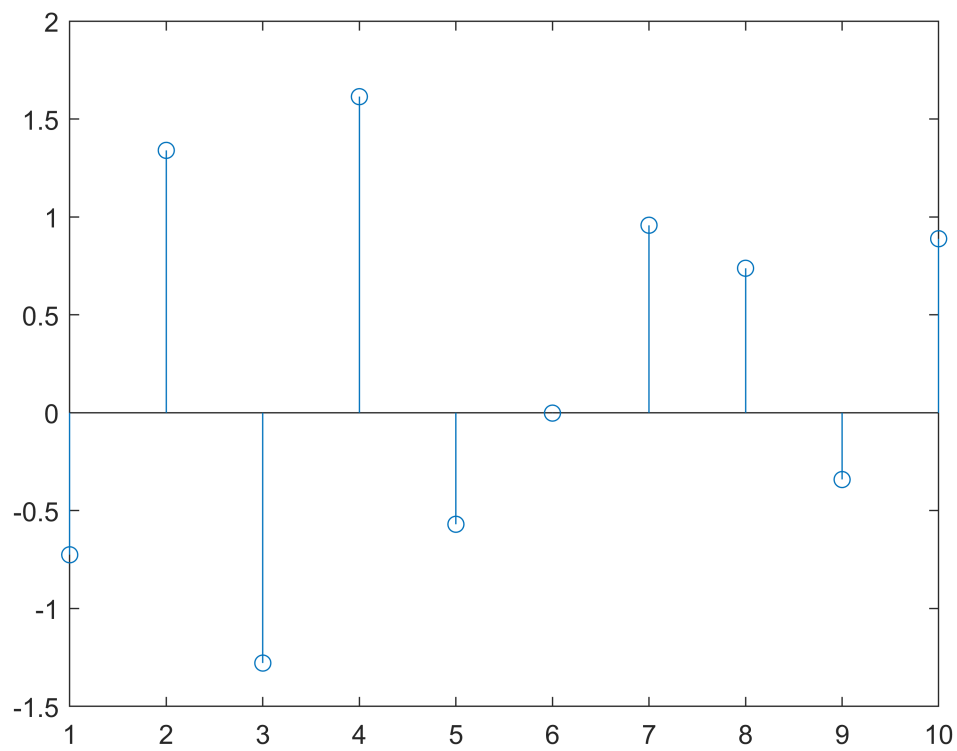
```
y=randn(10,1)
```

```
y = 10x1
-0.7257
 1.3399
-1.2787
 1.6141
-0.5698
-0.0020
 0.9575
 0.7378
-0.3414
 0.8886
```

```
bar(y)
```

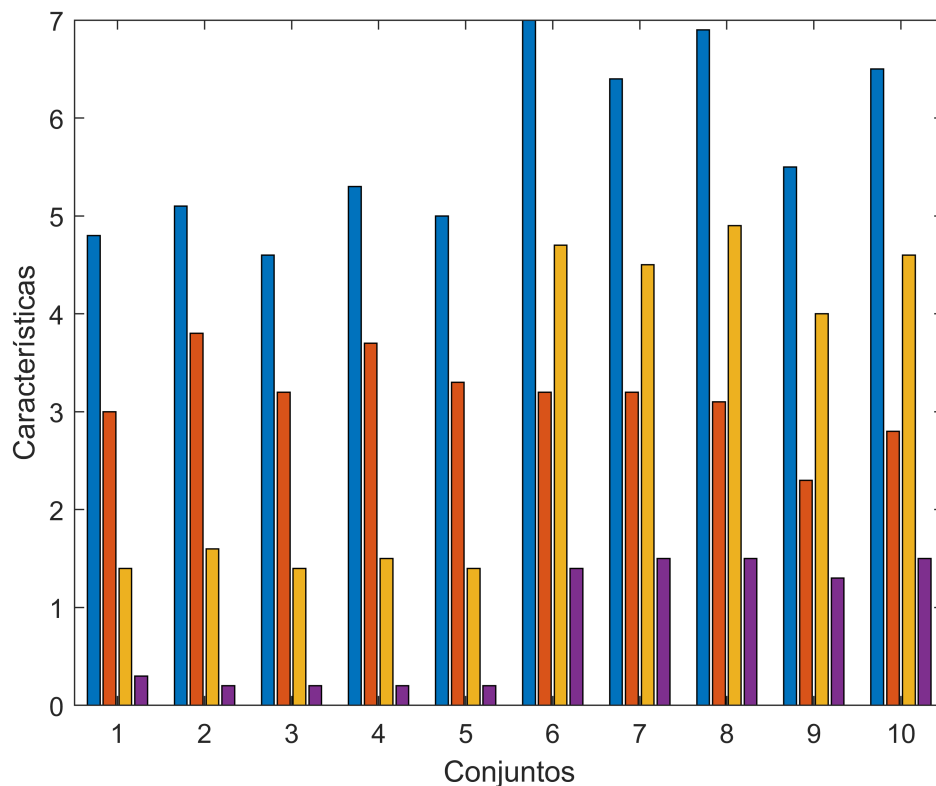


```
stem(y)
```



2.10 Dê os comandos abaixo e verifique se o gráfico de barras gerado permite diferenciar as espécies de flor setosa a versicolor. Ver [Fisheriris](#)

```
load fisheriris
bar(meas(46:55,:));
xlabel('Conjuntos');
ylabel('Características');
```



Resposta:

O gráfico de barras não permite diferenciar as espécies de flor setosa a versicolor.

Para diferenciar as espécies deve ser analisado essas informações em gráficos de dispersão, nos quais os será possível observar agrupamentos relevantes que permitem essa diferenciação.

Atividade 3: Comandos para definir e manipular modelos

3.1 Defina a função de transferência $G = \frac{10}{s^2 + (I/12) * s + 10}$ e obtenha sua sobrelevação e tempo de estabelecimento (Dica: stepinfo).

```
num =[10];
dem= [1 (I/12) 10];
G=tf(num,dem)
```

G =

```
      10
-----
s^2 + 0.08333 s + 10
```

Continuous-time transfer function.

S=stepinfo(G)

```
S = struct with fields:
    RiseTime: 0.3329
    SettlingTime: 93.4339
    SettlingMin: 0.0795
    SettlingMax: 1.9594
    Overshoot: 95.9447
    Undershoot: 0
    Peak: 1.9594
    PeakTime: 0.9935
```

3.2 Obtenha os polos e zeros de G.

pole(G)

```
ans = 2x1 complex
    -0.0417 + 3.1620i
    -0.0417 - 3.1620i
```

zero(G)

ans =

```
0x1 empty double column vector
```

3.3 Mostre a resposta de G a um degrau de amplitude I.

step(G)*I

```
ans = 1300x1
    0
    0.0488
    0.1899
    0.4089
```

```

0.6836
0.9870
1.2891
1.5605
1.7750
1.9123
:
:

```

3.4 Obtenha a função de transferência de malha fechada de G

```
M=feedback(G,1)
```

```
M =
```

```

      10
-----
s^2 + 0.08333 s + 20

```

```
Continuous-time transfer function.
```

3.5 Crie uma tabela onde cada linha corresponde a um ganho K e nas colunas sejam mostrada a sobrelevação, o tempo de subida e tempo de estabelecimento (ver itens 2.8 e 2.9 de [Matlab](#)) para o sistema de malha fechada com estes ganhos, $M(s) = \frac{KG(s)}{1 + KG(s)}$, para G(s) do item 3.1. Dica: table e stepinfo.

```

Swf1 = stepinfo(G* 1);
up1=Swf1.Overshoot;
ts1=Swf1.SettlingTime;
tr1=Swf1.RiseTime;

Swf2 = stepinfo(G* 10);
up2=Swf2.Overshoot;
ts2=Swf2.SettlingTime;
tr2=Swf2.RiseTime;

Swf3 = stepinfo(G* 100);
up3=Swf3.Overshoot;
ts3=Swf3.SettlingTime;
tr3=Swf3.RiseTime;

Swf4 = stepinfo(G* 1000);
up4=Swf4.Overshoot;
ts4=Swf4.SettlingTime;
tr4=Swf4.RiseTime;

Overshoot= [up1;up2;up3;up4];
SettlingTime = [ts1; ts2; ts3; ts4];

```

```
RiseTime = [tr1; tr2; tr3; tr4];
K=[1; 10; 100; 1000];
```

```
table(K, Overshoot, SettlingTime, RiseTime)
```

```
ans = 4x4 table
```

	K	Overshoot	SettlingTime	RiseTime
1	1	95.9447	93.4339	0.3329
2	10	95.9447	93.4339	0.3329
3	100	95.9447	93.4339	0.3329
4	1000	95.9447	93.4339	0.3329

Troque a figura ao lado por outra de sua preferência.

