



Matlab Programlamaya Giriş

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



MATLAB Yazılımı

- MATLAB **MAT**rix **LAB**oratory (Matris Laboratuvarı) anlamına gelmektedir.
- 1970'li yıllarda Cleve Barry Moler tarafında geliştirilmiştir.
- Cleve Barry Moler ve Jack Little 1984 yılında MathWorks Şirketini kurmuşlardır.
- Bugün, MATLAB yazılımı MathWorks Şirketi tarafından geliştirilmekte ve ticari olarak satılmaktadır.



Cleve Barry Moler



Jack Little

MATLAB

- MATLAB ile aşağıdaki işlemler mümkündür:
 - Sayısal hesaplamalar
 - Algoritma geliştirme
 - Veri analizi
 - Görselleştirme
- MATLAB'ın öğrenci versiyonu aşağıdaki linkten satın alınabilir:
- MATLAB and Simulink Student Suite (**\$55**)
- <https://www.mathworks.com/products/matlab/student.html>

MATLAB'ın Çalıştırılması

- Başlat menüsü → Tüm Programlar → MATLAB → R2013b → MATLAB R2013b
- Masaüstündeki MATLAB ikonuna çift tıklayarak
- Çalıştır ... → "matlab.exe" or "matlab" yazarak



MATLAB Penceresi

The screenshot displays the MATLAB R2013b interface. The top menu bar includes HOME, PLOTS, and APPS. Below it is a ribbon with various toolboxes like FILE, VARIABLE, CODE, ENVIRONMENT, and RESOURCES. The main workspace is divided into three panels:

- Current Folder:** Shows a file explorer view of the 'D:\matlab' directory. Files include folders like 'final', 'konu3', 'konu4', 'konu5', 'konu6', 'konu12', 'konu13', 'konu15', 'vize3' and files like 'asal.asv', 'asal.m', 'convert.asv', 'convert.m', 'imge1.bmp', 'logo.png', 'renk.bmp', 'renk.m', 'rgb2gray.m', 'topl.m', and 'ucgen.m'. The 'logo.png' file is selected.
- Command Window:** Contains MATLAB code and its output. The code defines a 5x5 magic square, sets variables 'a=2', 'b=5', and 'c=a+b'. The output shows the magic square and the values of 'a', 'b', and 'c'.

```
1 1 1 1 1
1 2 3 4 5
1 3 6 10 15
1 4 10 20 35
1 5 15 35 70

>> logo
>> a=2

a =

    2

>> b=5

b =

    5

>> c=a+b

c =

    7

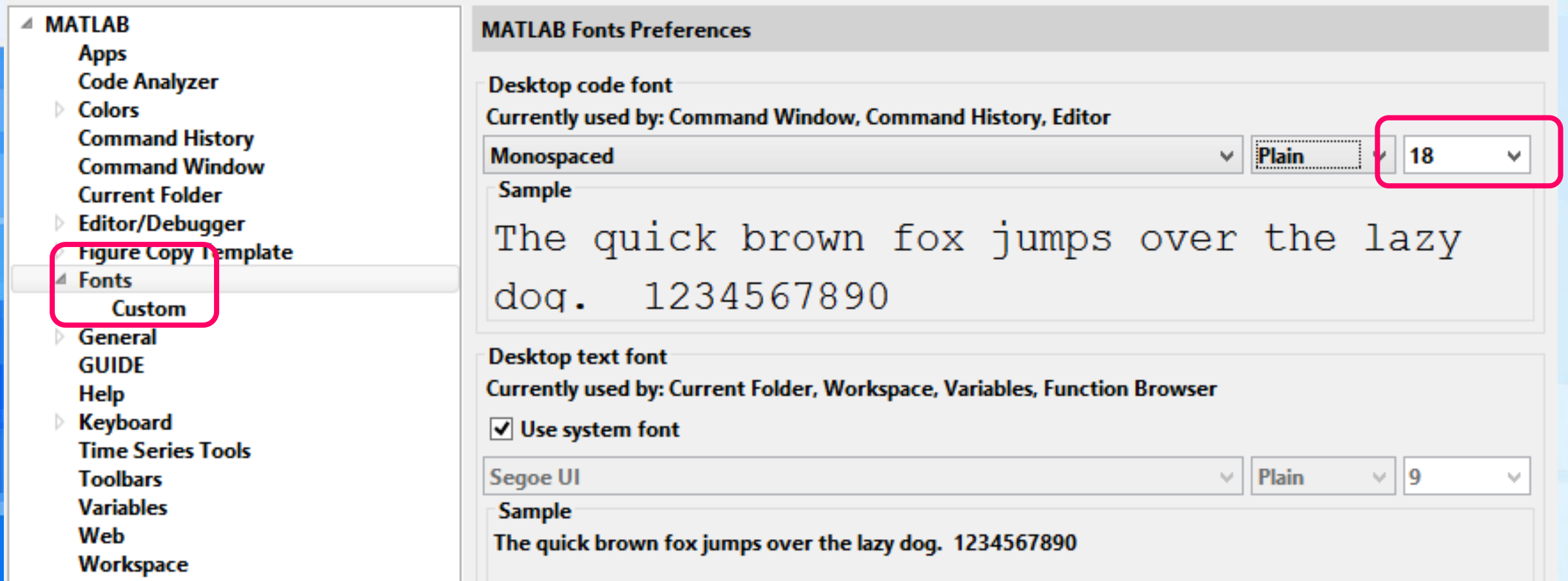
fx >>
```
- Workspace:** Lists variables in the current workspace. The 'Workspace' panel is highlighted with a black border and labeled 'Workspace'.

Name	Value
L	51x51 double
a	2
ans	5x5 double
b	5
c	7
I1	175.0011
I2	176.0011
logoFig	1
logoax	173.0011
s	174.0011
- Command History:** Shows a list of executed commands. The 'Command History' panel is highlighted with a green border and labeled 'Command History'.

```
07.12.2013 18:05 --
07.12.2013 19:18 --
    pascal(5)
    logo
    a=2
    b=5
    c=a+b
```

Yazı Boyutunun Ayarlanması

- Home sekmesi içinde yer alan  Preferences yolu ile tercihler açılır.



The image shows the MATLAB Preferences dialog box, specifically the 'Fonts' section. The 'Desktop code font' settings are highlighted with a red box. The font is set to 'Monospaced', the style is 'Plain', and the size is '18'. A sample text is displayed below the settings: 'The quick brown fox jumps over the lazy dog. 1234567890'. The 'Desktop text font' settings are also visible, showing 'Segoe UI' font, 'Plain' style, and size '9'. A sample text is displayed below: 'The quick brown fox jumps over the lazy dog. 1234567890'.

MATLAB

- Apps
- Code Analyzer
- Colors
- Command History
- Command Window
- Current Folder
- Editor/Debugger
- Figure Copy Template
- Fonts**
 - Custom
- General
- GUIDE
- Help
- Keyboard
- Time Series Tools
- Toolbars
- Variables
- Web
- Workspace

MATLAB Fonts Preferences

Desktop code font
Currently used by: Command Window, Command History, Editor

Monospaced | Plain | 18

Sample
The quick brown fox jumps over the lazy dog. 1234567890

Desktop text font
Currently used by: Current Folder, Workspace, Variables, Function Browser

Use system font

Segoe UI | Plain | 9

Sample
The quick brown fox jumps over the lazy dog. 1234567890

MATLAB'da Basit Komutlar

- Komut penceresinde $9*5$ yazarak Enter'a basın.
- Sonuç "ans" değişkenine atanır ve ekrana yazdırılır.
- ans değişkeninin değerini **Workspace**'den görmek mümkündür.
- Yapılan her farklı işlem sonucunda ans değişkeninin değeri değişir.

Workspace	
Name ▲	Value
ans	7

Command Window

```
>> 9*5
```

```
ans =
```

```
45
```

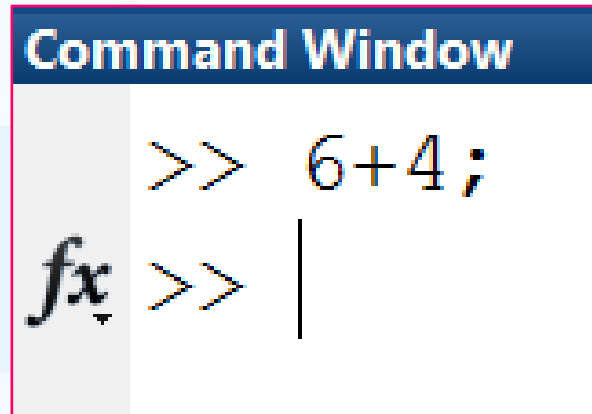
```
>> 28/4
```

```
ans =
```

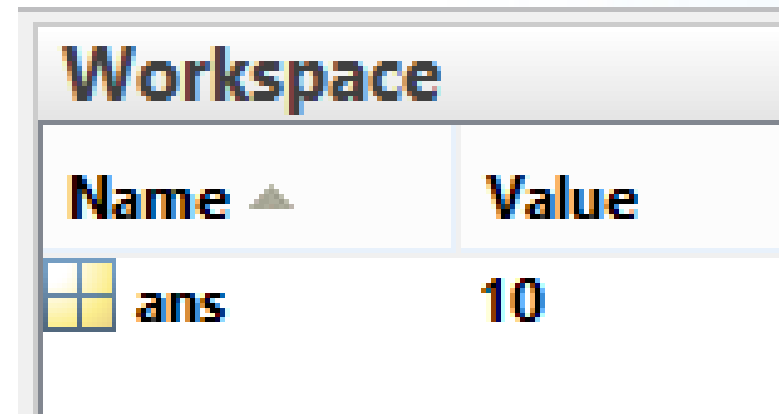
```
7
```


İşlem Sonucunun Ekrana Yazdırılmaması

- Matlab'da işlem sonucunun ekrana yazdırılmaması için konut sonuna ";" yazılır.
- İşlem sonucu Komut Penceresi'nde gösterilmez.
- Yeni hesaplanan değişken değeri Workspace'den görülebilir.



```
Command Window
>> 6+4;
fx >> |
```



Name ▲	Value
ans	10

Değişken Tanımlamak

- Aşağıdaki komutları yazınız:
- `a=5`
- `b=7`
- `c=a+b`
- Tanımlı tüm değişkenler ve değerleri, Workspace'den görülebilir.

```
>> a=5
```

```
a =
```

```
5
```

```
>> b=7
```

```
b =
```

```
7
```

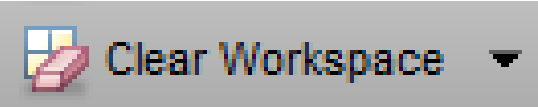
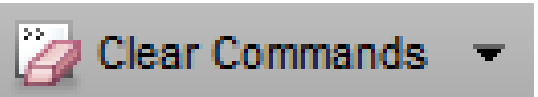
```
>> c=a+b
```

```
c =
```

```
12
```

Workspace	
Name ▲	Value
a	5
ans	7
b	7
c	12

"clear" ve "clc" Komutları

- "clear" komutu Workspace'de tanımlı tüm değişkenleri silmek için kullanılır.
- Alternatif olarak  düğmesi de kullanılabilir.
- `clear a b` a ve b değişkenlerini siler
- `clear x` x değişkenini siler.
- `clear all` tüm değişkenleri siler.
- "clc" Command Window (Komut penceresi içeriğini temizlemek için kullanılır.
- Alternatif olarak  düğmesi de kullanılabilir.

Matematiksel Fonksiyonlar

Fonksiyon	Açıklaması	Örnek
<code>rem(x, y)</code>	x'in y'ye bölümünden kalanı verir.	<code>rem(16,5)</code>
<code>sqrt(x)</code>	x'in karekökünü hesaplar.	<code>sqrt(64)</code>
<code>power(x, y)</code>	x'in y derece kuvvetini hesaplar. Veya x^y	<code>power(2,5)</code>
<code>round(x)</code>	x'i en yakın tam sayıya yuvarlar.	<code>round(3.67)</code>
<code>exp(x)</code>	e^x 'i hesaplar.	<code>exp(2)</code>
<code>log(x)</code>	x'in e tabanındaki logaritmasını hesaplar.	<code>log(3)</code>
<code>log10(x)</code>	x'in 10 tabanındaki logaritmasını hesaplar.	<code>log10(2)</code>
<code>abs(x)</code>	x'in tam değerini verir.	<code>abs(-5.9)</code>
<code>factorial(x)</code>	x'in faktöriyelini hesaplar.	<code>factorial(5)</code>

Trigonometrik Fonksiyonlar - 1

Fonksiyon	Açıklaması	Örnek
π	π 'nin değerini verir.	π
$\sin(x)$	x açısının sinüs değerini verir. (radyan)	$\sin(30)$
$\text{sind}(x)$	x açısının sinüs değerini verir. (derece)	$\text{sind}(30)$
$\text{asin}(x)$	x açısının arc sinüs değerini verir. (radyan)	$\text{asin}(45)$
$\text{asind}(x)$	x açısının arc sinüs değerini verir. (derece)	$\text{asind}(45)$
$\sinh(x)$	x açısının hiperbolik sinüs değerini verir. (radyan)	$\sinh(45)$
$\text{asinh}(x)$	x açısının hiperbolik sinüs değerini verir. (radyan)	$\text{asinh}(45)$

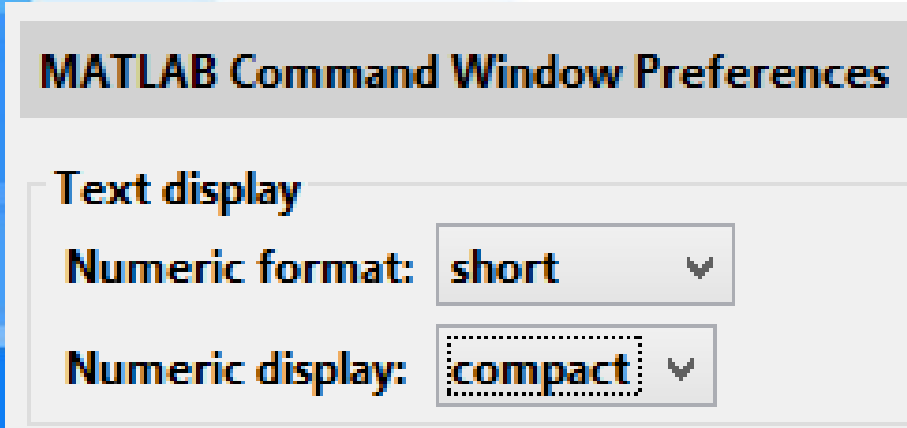
Trigonometrik Fonksiyonlar - 2

Fonksiyon	Açıklaması	Örnek
$\cos(x)$	x açısının kosinüs değerini verir. (radyan)	$\cos(45)$
$\tan(x)$	x açısının tanjant değerini verir. (radyan)	$\tan(45)$
$\cot(x)$	x açısının kotanjant değerini verir. (radyan)	$\cot(45)$
$\sec(x)$	x açısının sekant değerini verir. (radyan)	$\sec(45)$
$\csc(x)$	x açısının kosekant değerini verir. (radyan)	$\csc(45)$

Diğer fonksiyonlar da sinüs fonksiyonlarına benzerdir

Format Komutu

- `format short` ondalıklı basamakları 4 hane yazar
- `format long` ondalıklı basamakları 15 hane yazar
- `format loose` yazıları ayırık yazar
- `format compact` yazıları bitişik yazar



```
>> format long
>> pi
ans =
    3.141592653589793
>> format short
>> pi
ans =
    3.1416
```

```
>> format loose
>> pi

ans =

    3.1416

>> format compact
>> pi
ans =
    3.1416
```

Yardım Almak

- Bilgi edinilmek istenilen fonksiyonun adı **help** ile birlikte yazılır.
- Örnek: `help format`

```
>> help format
```

```
format Set output format.
```

```
format with no inputs sets the output format to t  
for the class of the variable. For float variable  
format SHORT.
```

```
format does not affect how MATLAB computations ar  
on float variables, namely single or double, are  
floating point precision, no matter how those var  
Computations on integer variables are done native  
variables are always displayed to the appropriate  
the class, for example, 3 digits to display the 1  
format SHORT and LONG do not affect the display c
```


Örnek Soru 1

- Bir dikdörtgen prizmanın kenar uzunlukları $a=5$, $b=6$ ve $c=8$ birimdir.
- Bu dikdörtgen prizmanın hacmini Matlab ile hesaplayınız.
- $V = a \times b \times c$
- Bu dikdörtgen prizmanın yanal alanını Matlab ile hesaplayınız.
- $A = 2 \times (a \times b + a \times c + b \times c)$
- Bu dikdörtgen prizmanın cisim köşegenini Matlab ile hesaplayınız.
- $K = \sqrt{a^2 + b^2 + c^2}$

Command Window

```
>> a=5;  
>> b=6;  
>> c=8;  
>> V=a*b*c  
V =  
    240  
>> A=2*(a*b+a*c+b*c)  
A =  
    236  
>> K=sqrt(a^2+b^2+c^2)  
K =  
    11.1803
```

```
>> K=sqrt(power(a,2)+power(b,2)+power(c,2))  
K =  
    11.1803
```

Örnek Soru 2

- Bir ABC üçgeninde iki kenarın uzunlukları şu şekilde verilmiştir: $a=4$, $b=5$.
- C açısının değeri 60° ise,
- Bu üçgenin alanını Matlab ile hesaplayınız.
- $A = \frac{1}{2} \times a \times b \times \sin(C)$

Command Window

```
>> a=4;  
>> b=5;  
>> A=0.5*a*b*sind(60)  
A =  
    8.6603
```

Örnek Soru 3

- $ax^2 + bx + c = 0$ şeklindeki 2. derece bir denklemin çözümünü Matlab ile yapınız.
- Örneğin: $x^2 - 2x - 15 = 0$
- $x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$, $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$

Örnek Soru 3 - Çözüm

Girdiler

```
Command Window
>> a=1
a =
    1
>> b=-2
b =
   -2
>> c=-15
c =
  -15
```

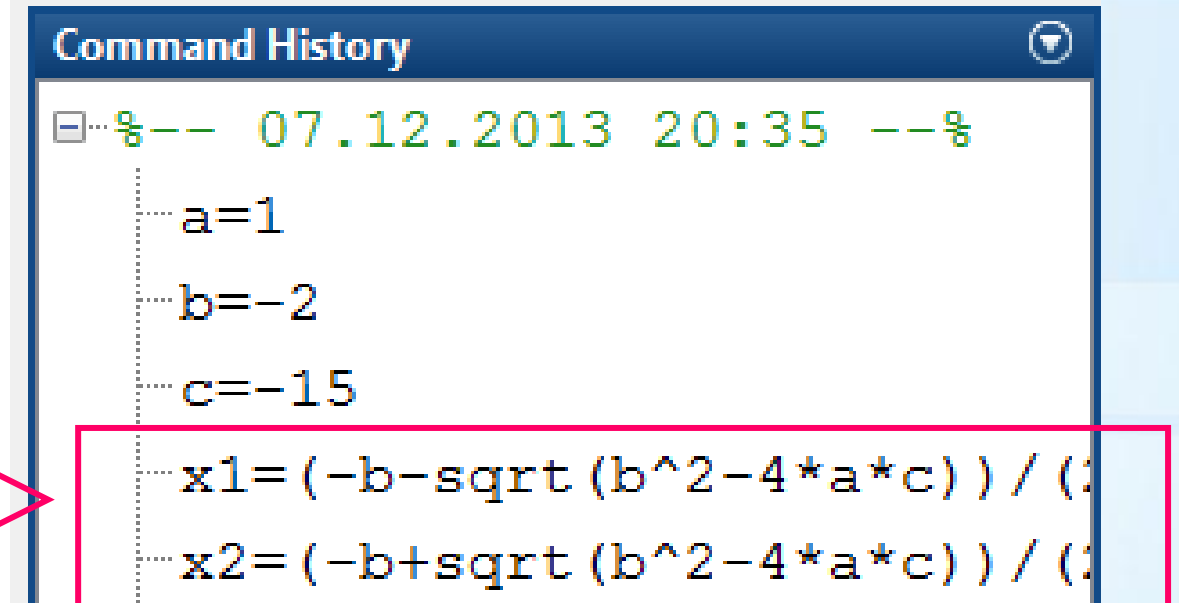
Çıktılar

```
>> x1=(-b-sqrt(b^2-4*a*c))/(2*a)
x1 =
   -3
>> x2=(-b+sqrt(b^2-4*a*c))/(2*a)
x2 =
    5
```

Workspace	
Name ▲	Value
a	1
b	-2
c	-15
x1	-3
x2	5

Command History Penceresi

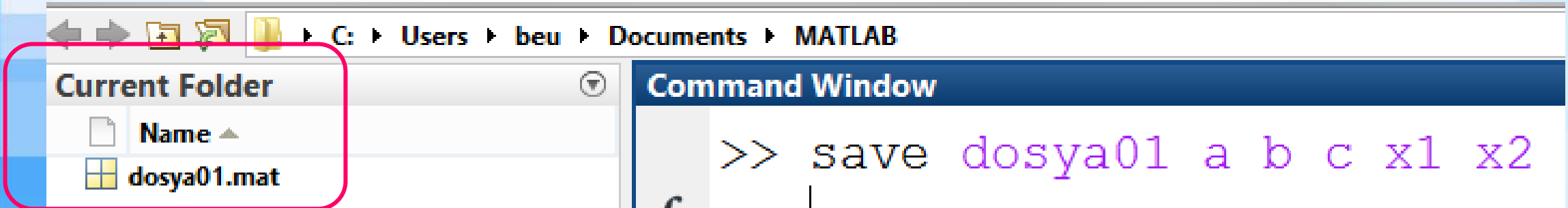
- Örnek Soru 3'deki 2.derece denklemi başka katsayılar ile çözmek için yeni a, b, ve c değerleri yazılır.
- x1 ve x2 değişkenlerini hesaplamak için, Komut Tarihçesi penceresindeki x1 ve x2 için yazılan ifadeler **çift tıklama** veya **sürükle bırak** yöntemleri ile tekrar çalıştırılır.
 - a=1
 - b=0
 - c=-4



```
Command History
---%-- 07.12.2013 20:35 --%
...a=1
...b=-2
...c=-15
...x1=(-b-sqrt(b^2-4*a*c))/(2*a)
...x2=(-b+sqrt(b^2-4*a*c))/(2*a)
```

Değişkenleri Kaydetmek ve Tekrar Yükleme

- Daha sonradan tekrar kullanılmak istenen değişkenler **save** komutu ile dosya adı verilerek **Current Folder** altında kaydedilir.
- `save dosya01 a b c x1 x2`



- Değişkenleri geri yüklemek için **load** komutu kullanılır.
- `load dosya01`
- Değişkenleri geri yüklemeden önce **clear** komutu ile değişkenleri silmeyi unutmayınız.



Matlab'da Matrisler

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Matlab'da Matrisler

➤ MATLAB'da tüm değişkenler matris formunda depolanır.

➤ $A = [5, 10, -2; -6, 0, 7]$

```
>> A=[5,10,-2;6,0,-7]
```

```
A =
```

```
     5     10     -2
```

```
     6      0     -7
```

➤ Aynı satırdaki elemanları ayırmak için `' , '` (virgül) veya `' '` (boşluk) kullanılır.

➤ Satırları birbirinden ayırmak için `' ; '` (noktalı virgül) kullanılır.

➤ Bir $A(m \times n)$ matrisinin boyutu:

- m: satır sayısı

- n: sütun sayısı

Özel Matrisler

➤ Boş matris

- $B = []$

- $C = \backslash \backslash$

➤ Skalar matris (boyut: 1×1)

- $D = 23$

- $E = -117$

- $F = 11.68$

➤ Bir matrisin skalar olup olmadığının kontrol edilmesi

- `isscalar(...)`

- Matris skalarsa `isscalar` fonksiyonu "1" değilse "0" cevabını verir.

Özel Matrisler

➤ **Vektör Matrisleri** (tek satır veya tek sütun matrisleri)

➤ Satır vektörü (boyut: $1 \times n$)

➤ $G = [3 \ 5 \ 7 \ 13]$ veya $G = [3, 5, 7, 13]$

```
>> G = [3 5 7 13]
```

```
G =
```

```
      3      5      7     13
```

➤ Sütun vektörü (boyut: $m \times 1$)

➤ $H = [1; 0; 6; -7; 8]$

```
>> H=[1;0;6;-7;8]
```

```
H =
```

```
      1
```

```
      0
```

```
      6
```

```
     -7
```

```
      8
```

Satır Matrisi Oluşturma

➤ Matris adı = başlangıç : artış : bitiş

➤ $A = 2 : 0.4 : 4$

```
>> A = 2 : 0.4 : 4
```

```
A =
```

```
2.0000    2.4000    2.8000    3.2000    3.6000    4.0000
```

➤ Matris adı = başlangıç : azalış : bitiş

➤ $B = 10 : -1 : 5$

```
>> B = 10 : -1 : 5
```

```
B =
```

```
10     9     8     7     6     5
```

Satır Matrisi Oluşturma

- `linspace(x1, x2)` : $x1$ ve $x2$ arasında 100 eşit parça ile ayrılmış vektör oluşturur.

```
>> a=linspace(5,10)
a =
Columns 1 through 5
    5.0000    5.0505    5.1010    5.1515    5.2020
Columns 6 through 10
    5.2525    5.3030    5.3535    5.4040    5.4545
```

Name	Value
a	1x100 double

- `linspace(x1, x2, N)` : $x1$ ve $x2$ arasında N eşit parça ile ayrılmış vektör oluşturur.

```
>> a=linspace(5,10,5)
a =
    5.0000    6.2500    7.5000    8.7500   10.0000
```

Matrisin Satır veya Sütun Vektörü Olduğunun Tespiti

- Matrisin satır vektörü olduğunun bulunması
- `isrow(A)`
- Fonksiyonu satır vektörü olduğunda "1" değilse "0" değerini verir.
- Matrisin sütun vektörü olduğunun bulunması
- `iscolumn(B)`
- Fonksiyonu sütun vektörü olduğunda "1" değilse "0" değerini verir.

Matrislerde İndeksleme

➤ $B = [3 \ 5 \ 7 \ 10]$

$B(1)$ $B(2)$ $B(3)$ $B(4)$

➤ $B(\text{end}) = 10$

- $A(3, 4)$ 3. satır, 4. sütun
- $A(1, :)$ 1. satırın tamamı
- $A(:, 2)$ 2. sütununun tamamı

```
>> A = magic(4)
```

A =

$A(:, 2)$

$A(1, :)$

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

$A(3, 4) = 1$

Matrislerde Doğrusal İndeksleme

➤ $A(11) = A(3,3) = 6$

```
>> A(11)
```

```
ans =
```

```
6
```

➤ $A(5) = A(1,2) = 2$

```
>> A(5)
```

```
ans =
```

```
2
```

İndeks
No

1 16	5 2	9 3	13 13
2 5	6 11	10 10	14 8
3 9	7 7	11 6	15 12
4 4	8 14	12 15	16 1

A

Matrislerde Doğrusal İndeksleme

A(1)	A(4)	A(7)
A(2)	A(5)	A(8)
A(3)	A(6)	A(9)

B(1)	B(6)	B(11)
B(2)	B(7)	B(12)
B(3)	B(8)	B(13)
B(4)	B(9)	B(14)
B(5)	B(10)	B(15)

Matrisler Hakkında Bilgi Veren Fonksiyonlar

- `numel()`
- Matristeki eleman sayısını verir.

```
>> numel(A)

ans =

    16
```

- `size()`
- Matrisin boyutlarını verir.

```
>> size(A)

ans =

     4     4
```

- `length()`
- Matrisin uzunluğunu verir.

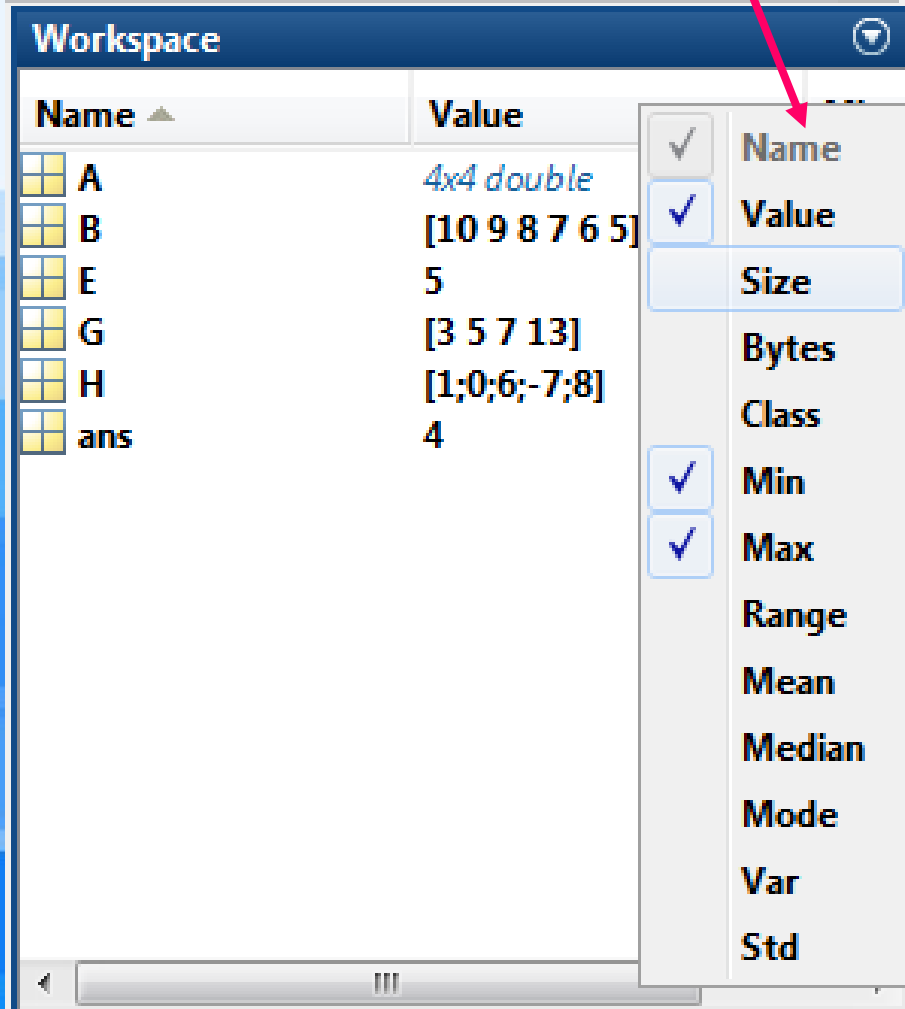
```
>> length(A)

ans =

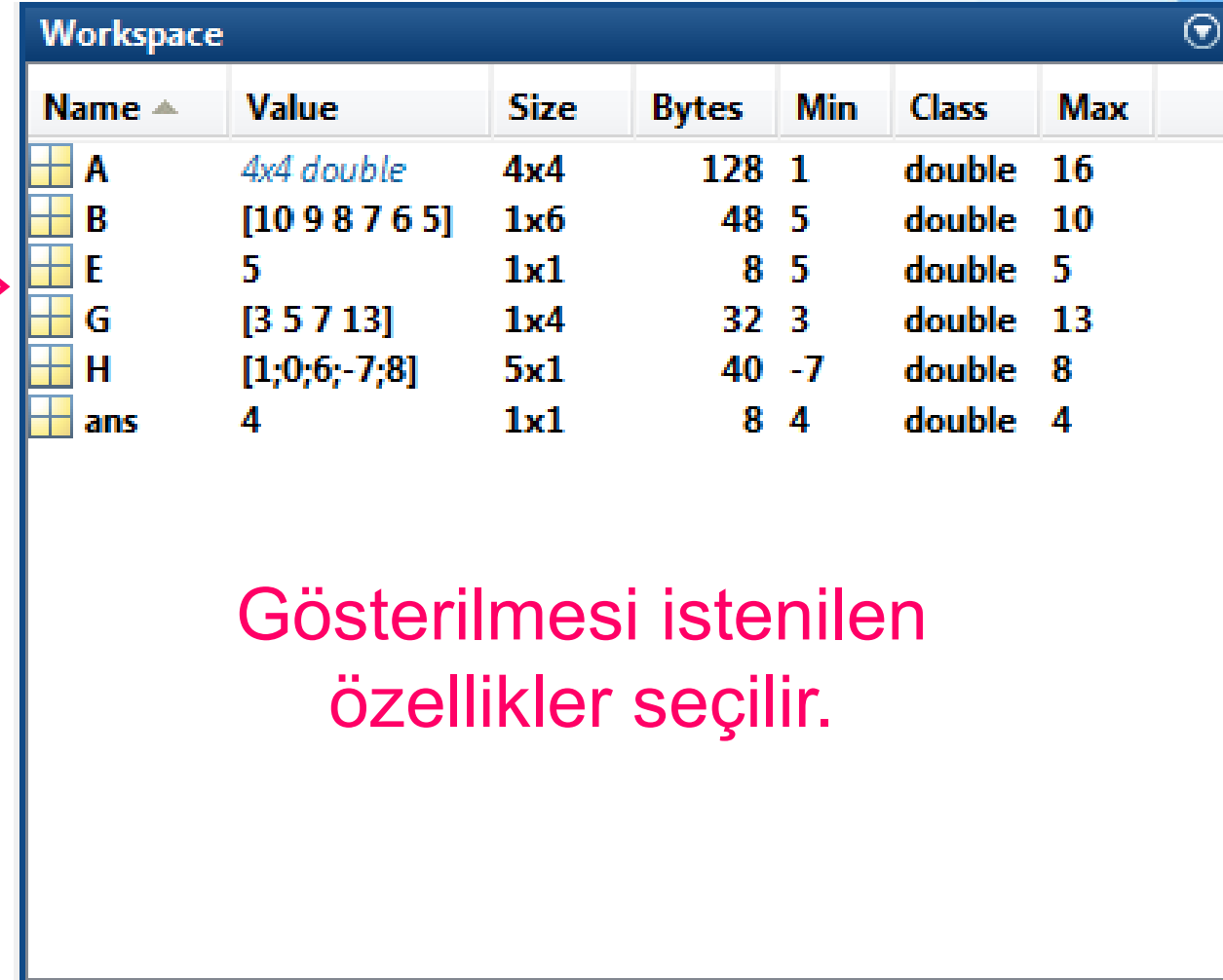
     4
```

Workspace Görünümünü Deęiřtirmek

Bu alana saę tuř ile tıklanır.



Name ▲	Value
A	4x4 double
B	[10 9 8 7 6 5]
E	5
G	[3 5 7 13]
H	[1;0;6;-7;8]
ans	4



Name ▲	Value	Size	Bytes	Min	Class	Max
A	4x4 double	4x4	128	1	double	16
B	[10 9 8 7 6 5]	1x6	48	5	double	10
E	5	1x1	8	5	double	5
G	[3 5 7 13]	1x4	32	3	double	13
H	[1;0;6;-7;8]	5x1	40	-7	double	8
ans	4	1x1	8	4	double	4

Gösterilmesi istenilen özellikler seçilir.

Matrisin Elemanlarının Silinmesi

- $B=[2, 4, 6, 8, 10]$ matrisinin 3. elemanını silmek isteyelim:
- $B(3) = []$ komutu ile üçüncü eleman silinir. Matris 4 elemanlı hale gelir.

```
>> B=[2, 4, 6, 8, 10];  
>> B(3) = []  
  
B =  
     2     4     8    10
```

- $C(:, 2) = []$ C matrisinde 2. sütunu siler.
- $D(1, :) = []$ D matrisinde 1. sütunu siler.

Özel Matris Oluşturma Fonksiyonları

- `zeros(m, n)` : tüm değerleri **0** olan matris oluşturur.

```
>> zeros(2,4)

ans =

     0     0     0     0
     0     0     0     0
```

- `ones(m, n)` : tüm değerleri **1** olan matris oluşturur.

```
>> ones(1,5)

ans =

     1     1     1     1     1
```

- `eye(m, n)` : **birim** matris oluşturur.

```
>> eye(3,4)

ans =

     1     0     0     0
     0     1     0     0
     0     0     1     0
```


Özel Matris Oluşturma Fonksiyonları

- `rand(m, n)` : uniform dağılımlı rastgele matris oluşturur.

```
>> rand(1,6)

ans =

    0.6557    0.0357    0.8491    0.9340    0.6787    0.7577
```

- `randn(m, n)` : normal dağılımlı rastgele matris oluşturur.

```
>> randn(2,5)

ans =

    0.7269    0.2939    0.8884   -1.0689   -2.9443
   -0.3034   -0.7873   -1.1471   -0.8095    1.4384
```

Matris İşlemleri

- + : toplama
- - : çıkarma
- * : çarpma

```
>> C=[1 2 3; 4 5 6; 7 8 9]
```

```
C =
```

```
     1     2     3
     4     5     6
     7     8     9
```

```
>> D= [10 11 12;13 14 15; 16 17 18]
```

```
D =
```

```
    10    11    12
    13    14    15
    16    17    18
```

```
>> E=C+D
```

```
E =
```

```
    11    13    15
    17    19    21
    23    25    27
```

```
>> E=D-C
```

```
E =
```

```
     9     9     9
     9     9     9
     9     9     9
```

```
>> E=C*D
```

```
E =
```

```
     84     90     96
    201    216    231
    318    342    366
```

Array (Dizi) İşlemleri

- Dizi (Element by element) işlemleri
- `.` * : element-by-element çarpım
- `.` / : element-by-element bölüm
- `.` ^ : element-by-element üs

```
>> F=[1,2;3,4]
F =
     1     2
     3     4
>> G=[2,4;6,8]
G =
     2     4
     6     8
```

```
>> H=F.*G
```

```
H =
```

```
     2     8
    18    32
```

```
>> H=G./F
```

```
H =
```

```
     2     2
     2     2
```

```
>> H=F.^2
```

```
H =
```

```
     1     4
     9    16
```

Matrisin Transpozuzu

- `'` : matrisin transpozunu hesaplar.

```
A =  
  
     6     0    -7  
     2     5    -8  
  
>> A'  
  
ans =  
  
     6     2  
     0     5  
    -7    -8
```

Kare Matrisin Determinantı

- `det ()` : kare matrisin determinantını verir.

```
C =  
  
     1     2     3  
     4     5     6  
     7     8     0  
  
>> det (C)  
  
ans =  
  
    27.0000
```

Matrisin Tersisi

- `inv()` : Matrisin tersini verir.
- Bir matrisin kendisi ile tersinin çarpımı birim matristir.

```
C =  
  
     1     2     3  
     4     5     6  
     7     8     0  
  
>> D=inv(C)  
  
D =  
  
    -1.7778    0.8889   -0.1111  
     1.5556   -0.7778    0.2222  
    -0.1111    0.2222   -0.1111  
  
>> E=C*D  
  
E =  
  
     1.0000         0   -0.0000  
    -0.0000     1.0000         0  
     0.0000   -0.0000     1.0000
```

Linear Denklem Sistemi Çözümleri

➤ Aşağıdaki şekilde lineer denklem sistemi varsa,

- $2x - y + z = 8$

- $-x + y - 2z = -9$

- $x + y - 3z = -8$

➤ $A = \begin{bmatrix} 2 & -1 & 1 \\ -1 & 1 & -2 \\ 1 & 1 & -3 \end{bmatrix}$

➤ $b = \begin{bmatrix} 8 \\ -9 \\ -8 \end{bmatrix}$ olarak tanımlanır.

➤ Denklem sisteminin çözümü: $Ac=b$

➤ Çözüm: $c=A \setminus b$

- $c = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

```
>> A=[2 -1 1; -1 1 -2; 1 1 -3];  
>> b=[8;-9;-8];  
>> c=A\b  
  
c =  
  
    2.0000  
   -1.0000  
    3.0000
```

Örnek Soru 1

- `A=magic(5)` komutu ile bir matris oluşturunuz.
- Aşağıdaki ifadelerin cevaplarını komutları yazmadan vermeye çalışınız. Daha sonra komutları yazarak cevaplarınızı kontrol ediniz.

- `A(8)` =?
- `A(3,4)` =?
- `A(19)` =?
- `A(2,:)` =?
- `A(:,3)` =?

```
>> A=magic(5)
```

```
A =
```

```
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```


Örnek Soru 2

- Aşağıdaki problemi Matlab'da lineer denklem sistemi yardımıyla çözünüz.
- Problem: Bir defter ile bir kalemin fiyatının toplamı 1 lira 10 kuruştur. Defterin fiyatı kalemin fiyatından 1 lira fazla olduğuna göre kalemin ve defterin fiyatlarını bulunuz.

Örnek Soru 2 - Çözüm

➤ Denklem Sistemi:

➤ $x + y = 1.10$

➤ $x - y = 1.00$

➤ Matlab'da çözüm:

➤ $A = [1, 1 ; 1, -1];$

➤ $b = [1.10; 1.00];$

➤ $c = A \setminus b$

➤ Defter = 1.05 TL, Kalem = 0.05 TL

```
>> A=[1, 1 ; 1, -1];
```

```
>> b=[1.10;1.00];
```

```
>> c=A\b
```

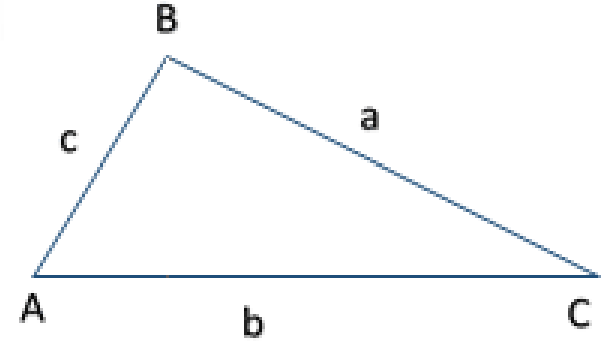
```
c =
```

```
1.0500
```

```
0.0500
```

Örnek Soru 3

- Bir çeşitkenar üçgenin alanı yarı çevre kullanılarak aşağıdaki formül ile hesaplanır.
- $u = \frac{(a+b+c)}{2}$, a, b ve c kenar uzunlukları
- $Alan = \sqrt{u \cdot (u - a) \cdot (u - b) \cdot (u - c)}$
- Matlab'da 5 elemanlı bir satır vektörü oluşturunuz (adı B olsun).
- Bu vektörün ilk 3 değerine üçgenin kenar uzunluklarını giriniz (5, 6 ve 7 birim).
- Vektörün 4. elemanın değeri olarak yarı çevreyi ve 5. eleman değeri olarak da üçgen alanını hesaplayınız.



Örnek Soru 3 - Çözüm

```
>> B=zeros(1,5)
```

```
B =
```

```
    0    0    0    0    0
```

```
>> B(1:3)=[5,6,7]
```

```
B =
```

```
    5    6    7    0    0
```

```
>> B(4)=(B(1)+B(2)+B(3))/2
```

```
B =
```

```
    5    6    7    9    0
```

```
>> B(5)=sqrt(B(4)*(B(4)-B(1))*(B(4)-B(2))*(B(4)-B(3)))
```

```
B =
```

```
    5.0000    6.0000    7.0000    9.0000   14.6969
```



Matlab'da İstatistiksel Fonksiyonlar ve Grafik Çizimi

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Temel İstatistiksel Fonksiyonlar

- `mean (A)` : A matrisinin aritmetik ortalamasını hesaplar
- `max (A)` : A matrisindeki maksimum değeri verir
- `min (A)` : A matrisindeki minimum değeri verir
- `mode (A)` : A matrisindeki en çok tekrarlayan veriyi gösterir
- `sum (A)` : A matrisinin tüm değerlerini toplar
- `sort (A)` : A matrisinin verilerini sıralar
- `median (A)` : A matrisinin medyan (ortanca) değerini verir
- `std (A)` : A matrisinin standart sapmasını hesaplar
- `range (A)` : A matrisinin aralığını (maks. - min) hesaplar

Temel İstatistiksel Fonksiyonlar

```
>> A=rand(1,6)
```

```
A =  
  
    0.6948    0.3171    0.9502    0.0344    0.4387    0.3816
```

```
>> sort(A)
```

```
ans =  
  
    0.0344    0.3171    0.3816    0.4387    0.6948    0.9502
```

```
>> mean(A)
```

```
ans =  
  
    0.4695
```

```
>> max(A)
```

```
ans =  
  
    0.9502
```

```
>> min(A)
```

```
ans =  
  
    0.0344
```

```
>> sum(A)
```

```
ans =  
  
    2.8169
```

```
>> median(A)
```

```
ans =  
  
    0.4102
```

```
>> std(A)
```

```
ans =  
  
    0.3172
```

Örnek Uygulama - 1

- `load count.dat` komutu ile örnek veri yükleyelim.
- `count` matrisindeki her bir sütunun ortalamasını ve standart sapmasını hesaplayalım.
- `count` matrisinin tümü için minimum ve maksimum değerleri hesaplayalım.

```
>> load count.dat
>> mean(count)
ans =
    32.0000    46.5417    65.5833
>> std(count)
ans =
    25.3703    41.4057    68.0281
>> min(count(:))
ans =
     7
>> max(count(:))
ans =
    257
```


Örnek Uygulama - 2

- `load count.dat` komutu ile örnek veri yükleyelim.
- `count` matrisinin ilk sütunu A vektörüne, ikinci sütunu B vektörüne ve üçüncü sütunu C vektörüne eşitlensin.
- Ardından sonraki slayttaki hesapları yapınız.

```
>> load count.dat  
>> A=count(:,1);  
>> B=count(:,2);  
>> C=count(:,3);
```

Örnek Uygulama - 2

- A vektörünün ortalama, maksimum, minimum ve aralığını hesaplayınız.
- B vektöründeki tüm sayıları toplayınız. B veri setindeki modu bulunuz.

```
>> mean(A)
ans =
    32
>> min(A)
ans =
     7
>> max(A)
ans =
   114
>> range(A)
ans =
   107
```

```
>> sum(B)
ans =
   1117
>> mode(B)
ans =
     9
```

Örnek Uygulama - 2

- C vektörünün ortalamasını, standart sapmasını ve medyanını hesaplayınız.

```
>> C=count (:, 3);  
>> mean (C)  
ans =  
    65.5833  
>> std (C)  
ans =  
    68.0281  
>> median (C)  
ans =  
    39
```

Örnek Uygulama - 2

- C vektörünü yazdırınız.
Ardından C vektöründeki değerleri sıralı olarak D ve E (artan ve azalan sıralı) vektörlerine eşitleyiniz.
- `D=sort(C)` veya
`D=sort(C, 'ascend')`
- `D=sort(C, 'descend')`

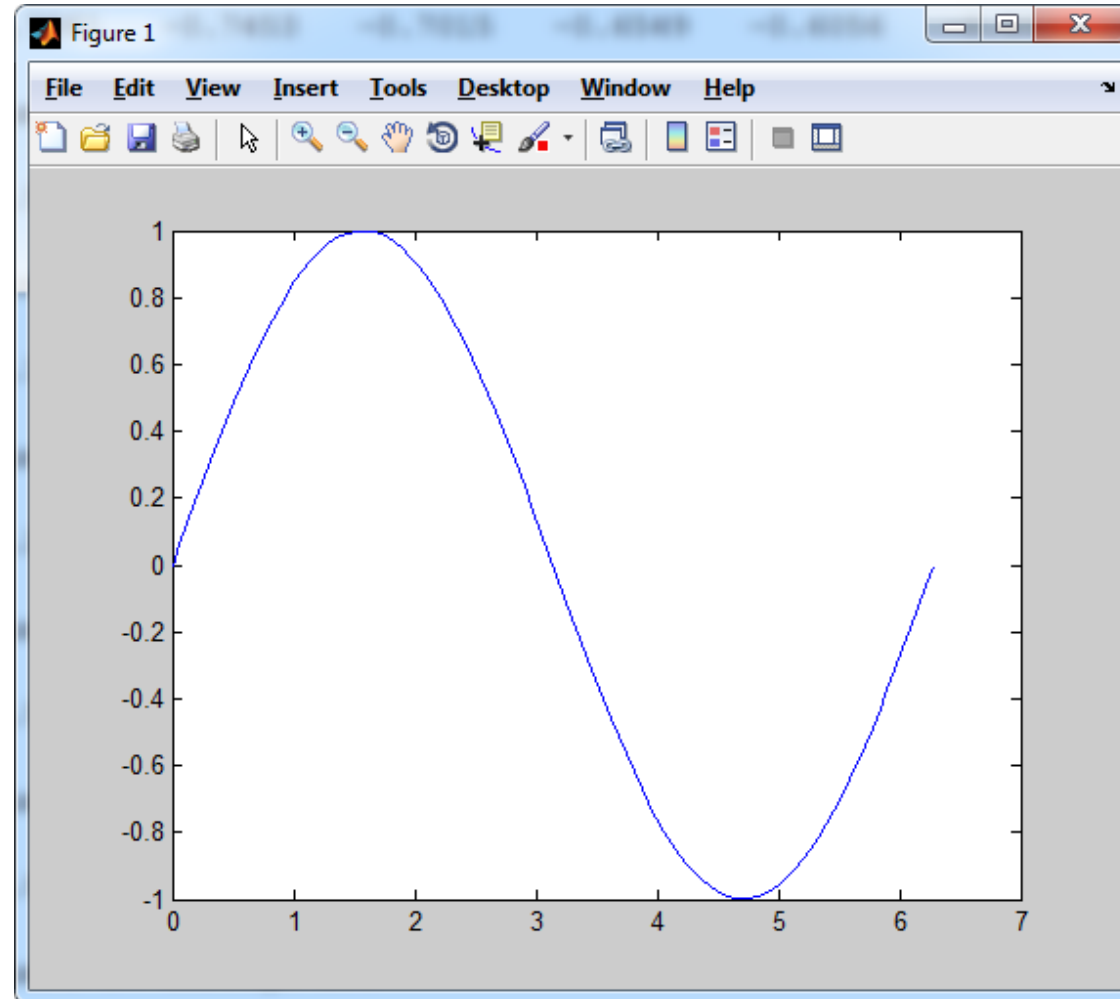
```
>> D=sort(C)
D =
     7
     9
     9
    10
    11
    14
    15
    15
    18
    20
    29
    30
    48
```

```
>> E=sort(C, 'descend')
E =
   257
   186
   180
   151
   115
    92
    90
    76
    69
    68
    55
    48
```

MATLAB ile Grafik Çizimi

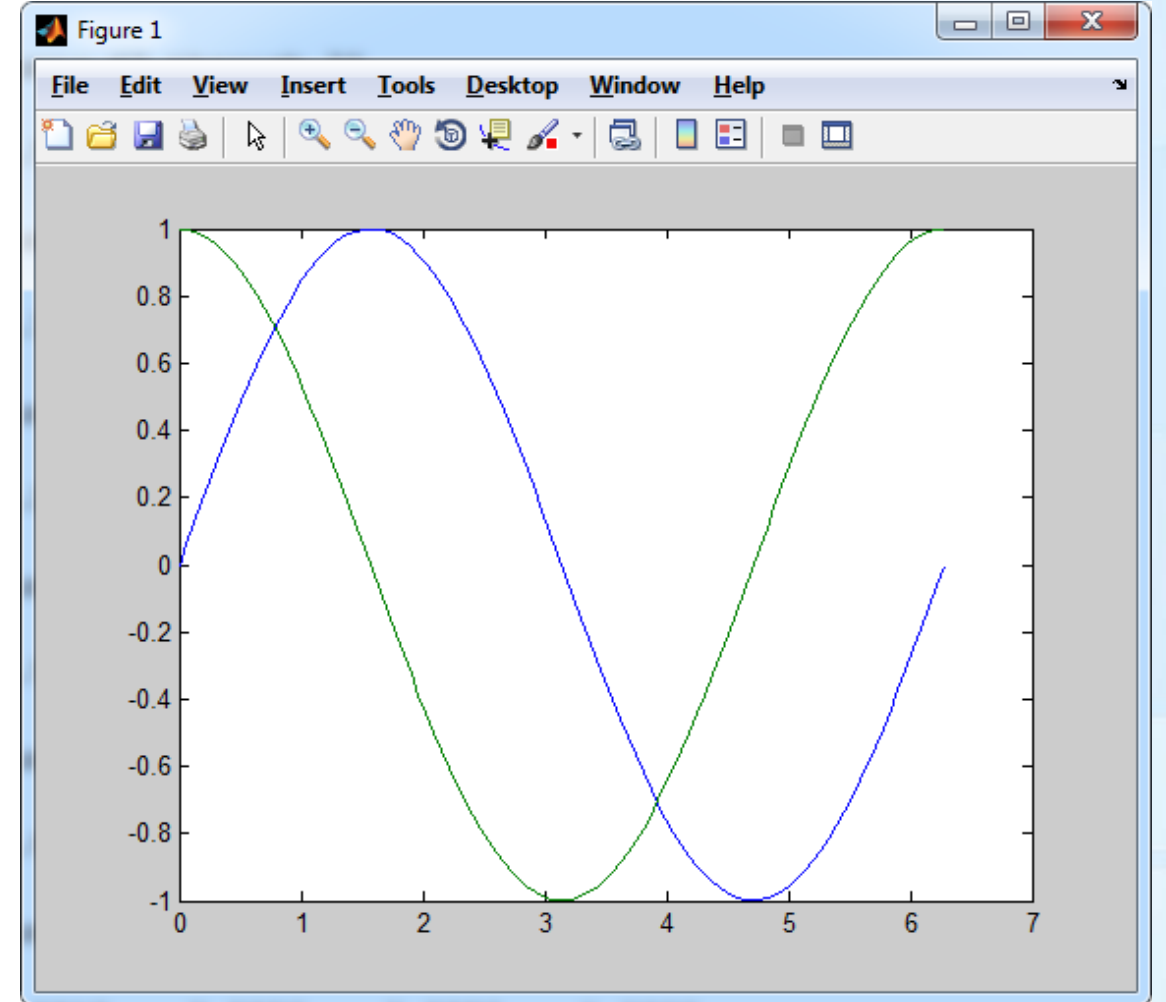
- `linspace(x1, x2, n)` komutu ile grafik için veri oluşturalım.
- `x=linspace(0, 2*pi, 100)`
- `y=sin(x)`
- `plot(x, y)`
- Plot fonksiyonu x ve y değerlerinin grafiğini çizer.

plot(x,y) Komutu Sonucu



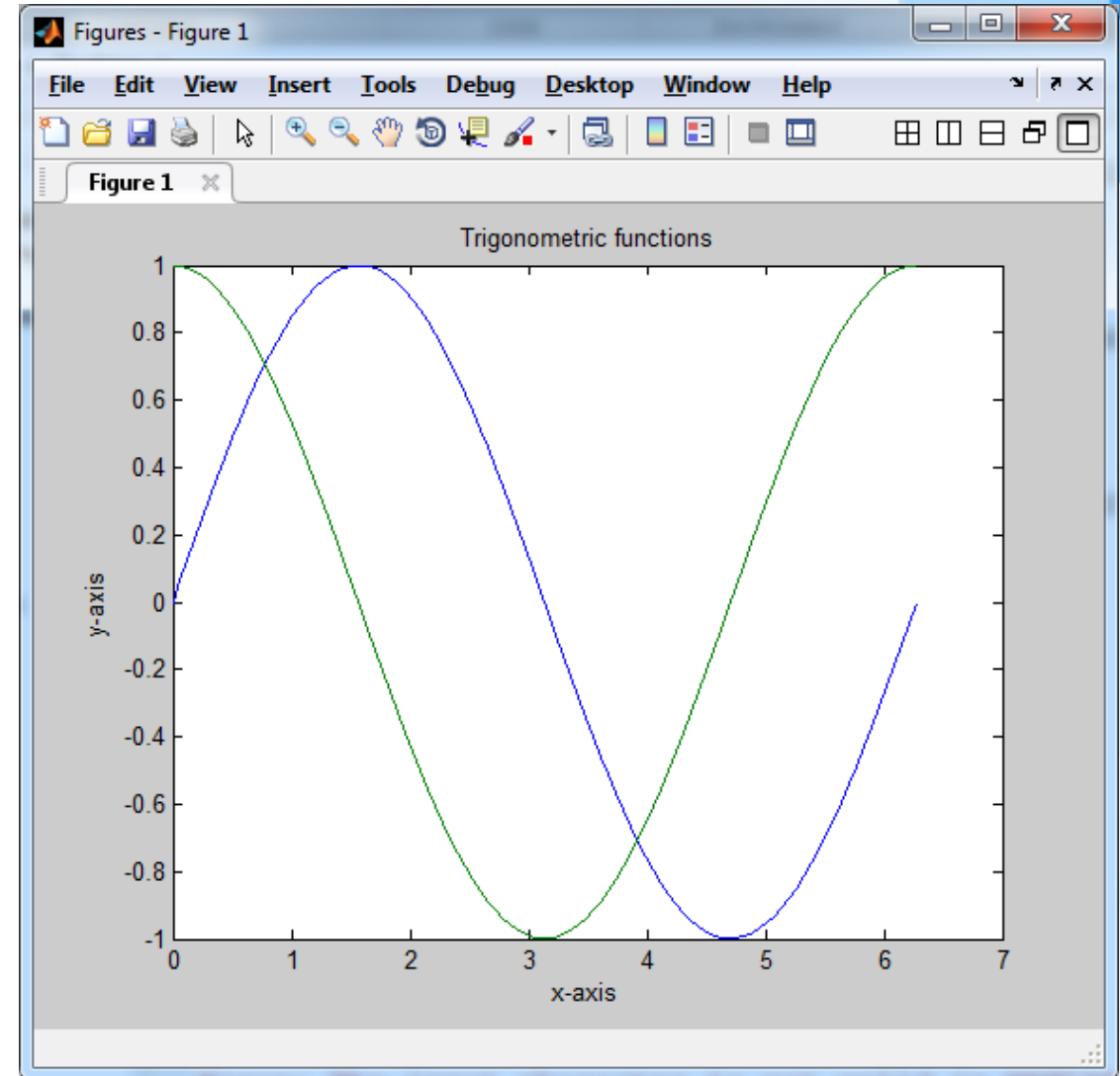
MATLAB ile Grafik Çizimi

- $z = \cos(x)$
- `plot(x, y, x, z)`
- **x-y ile x-z veri çiftlerini aynı grafik üzerinde gösterir.**



Grafik ve Eksen Başlıkları

- Title komutu ile grafik başlığı yazılır.
- `title('Trigonometric functions');`
- `xlabel` ve `ylabel` komutları ile eksen başlıkları yazılır.
- `xlabel('x-axis');`
- `ylabel('y-axis');`



help plot ile Grafik Yardımı

```
>> help plot
```

```
plot Linear plot.
```

```
plot(X,Y) plots vector Y versus vector X. If X or Y is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up. If X is a scalar and Y is a vector, disconnected line objects are created and plotted as discrete points vertically at X.
```

```
plot(Y) plots the columns of Y versus their index. If Y is complex, plot(Y) is equivalent to plot(real(Y),imag(Y)). In all other uses of plot, the imaginary part is ignored.
```

```
Various line types, plot symbols and colors may be obtained with plot(X,Y,S) where S is a character string made from one element from any or all the following 3 columns:
```

b	blue
g	green
r	red
c	cyan
m	magenta
y	yellow
k	black
w	white

Çizgi renkleri

Nokta Stilleri

.	point
o	circle
x	x-mark
+	plus
*	star
s	square
d	diamond
v	triangle (down)
^	triangle (up)
<	triangle (left)
>	triangle (right)
p	pentagram
h	hexagram

-	solid
:	dotted
-.	dashdot
--	dashed
(none)	no line

Çizgi Stilleri

Stilleri Kullanarak Grafik Çizmek

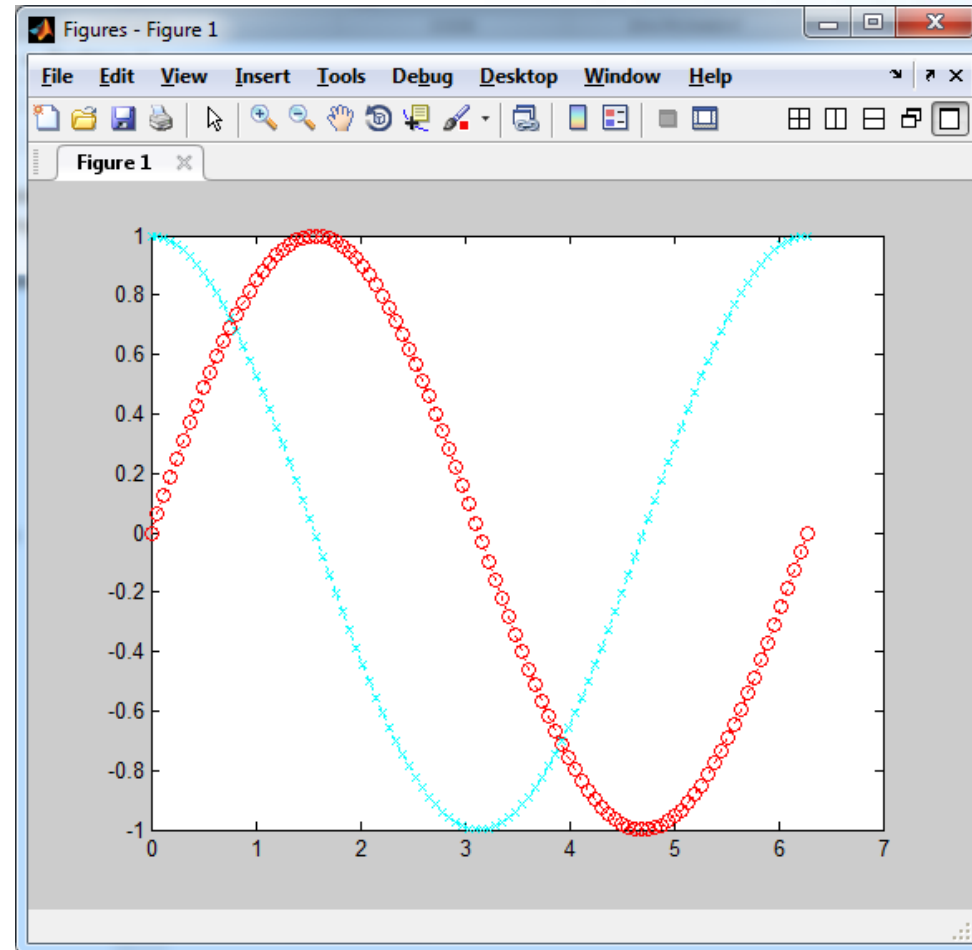
➤ `plot(x, y, 'r:o', x, z, 'c:x')`

➤ Ya da

➤ `plot(x, y, 'r:o')`

➤ `hold`

➤ `plot(x, z, 'c:x')`

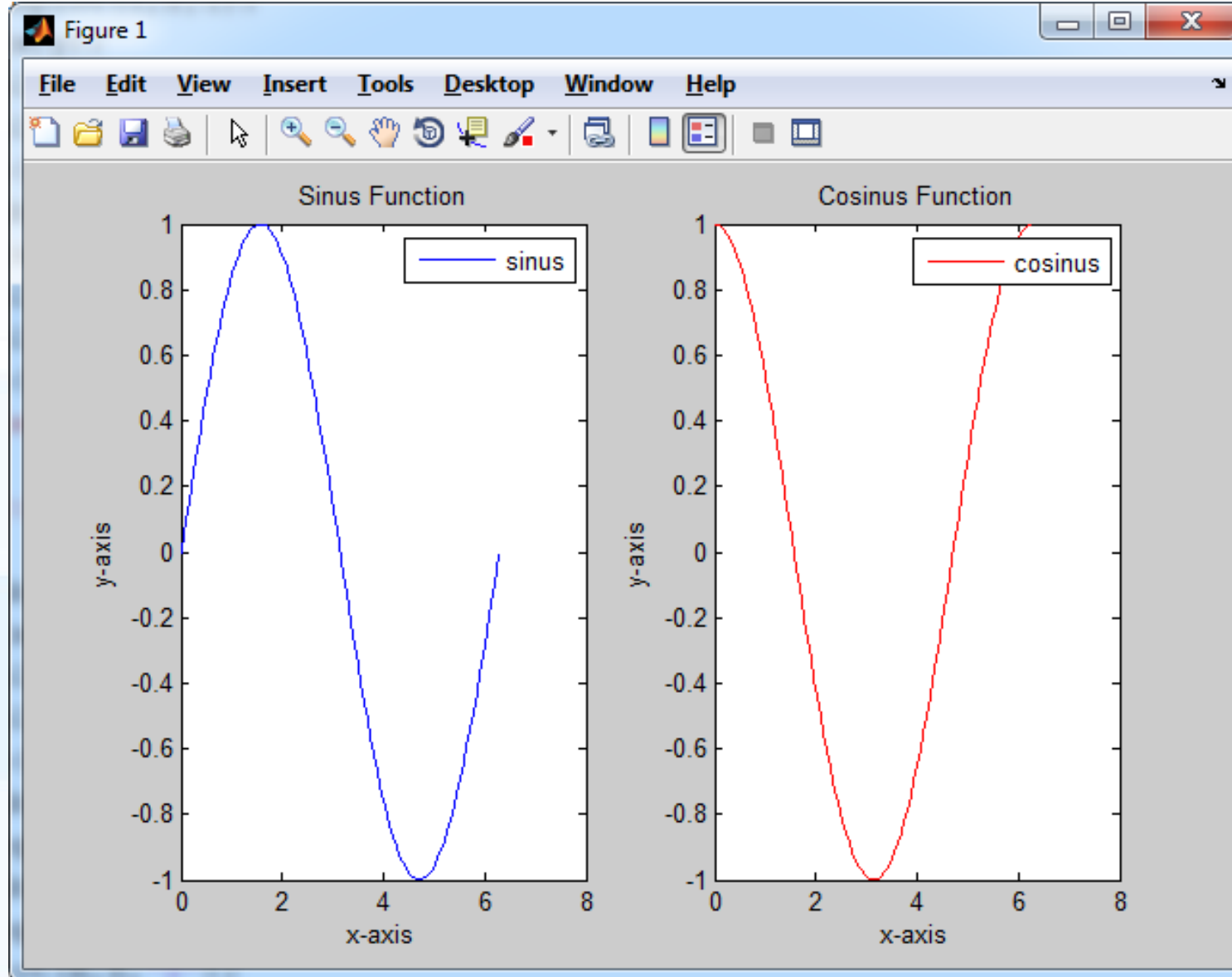


Subplot Komutu ile Çoklu Grafik Çizimi

- `subplot(1,2,1);`
- `plot(x,y);`
- `xlabel('x-axis');`
- `ylabel('y-axis');`
- `legend('sinus');`
- `title('Sinus Function')`

- `subplot(1,2,2);`
- `plot(x,z,'r');`
- `ylabel('y-axis');`
- `xlabel('x-axis');`
- `legend('cosinus');`
- `title('Cosinus Function')`

Subplot Komutu ile Çoklu Grafik Çizimi



Eksen Komutları

- `Axis` komutu ile eksenlerin başlangıç ve bitiş değerleri ayarlanır.
- `axis([xmin xmax ymin ymax])`
- Eksenleri gizlemek ve göstermek
 - `axis off` :Ekseni gizler
 - `axis on` :Ekseni gösterir

Property Editor ile Grafiğin Düzenlenmesi

View => Property Editor

The screenshot shows the MATLAB interface with the 'View' menu open, highlighting the 'Property Editor' option. Below the menu, two plots are visible: a blue sine wave and a red cosine wave. The 'Property Editor - Axes' window is open at the bottom, showing the title 'Cosinus Function' and various font settings for the X-axis, Y-axis, Z-axis, and Font. The Font settings include: Font Name: Helvetica, Font Size: 10.0, Weight: Normal, and Angle: Normal. The X-axis has a title 'sinus' and the Y-axis has a title 'y-axis'. The cosine plot has a title 'Cosinus Function' and the Y-axis has a title 'y-axis'.

Figure 1: Property Editor - Axes

Title: Cosinus Function

Colors: [Color Selection] [Color Selection]

Grid: X Y Z Box

X Axis Y Axis Z Axis Font

Font Name: Helvetica

Font Size: 10.0

Weight: Normal

Angle: Normal

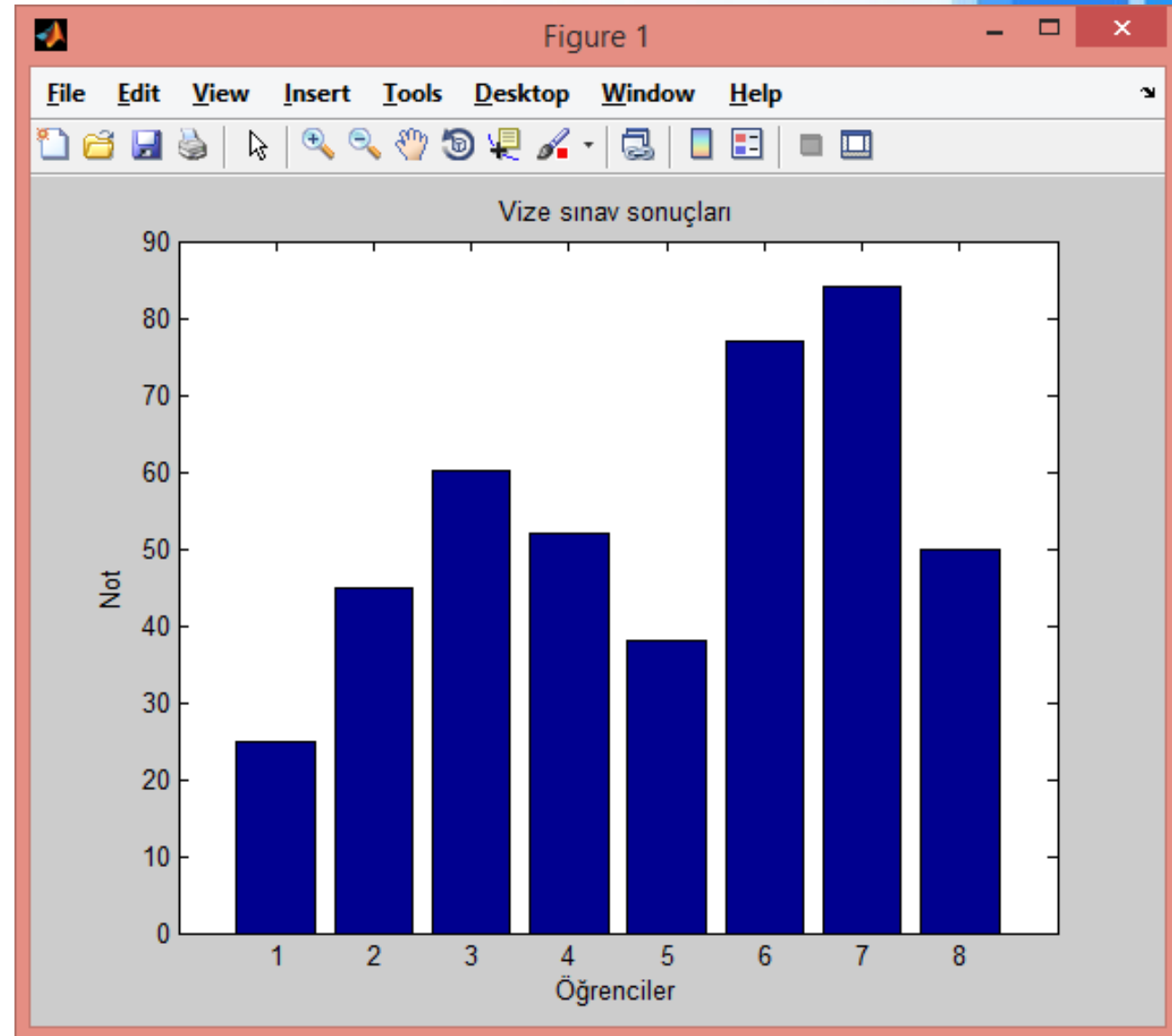
More Properties...

Bar Grafiği

- 8 tane öğrencinin sınav notunu gösteren bar grafiği çizmek için öğrenci sayısını gösteren x ve notları gösteren y matrislerini oluşturalım:
- `x=[1:8];`
- `y=[25 45 60 52 38 77 84 50];`
- `bar(x,y)` ile bar grafiği çizilir. Eksen ve grafik başlıkları aynı şekilde yazılır.
- `xlabel('Öğrenciler');`
- `ylabel('Not');`
- `title('Vize sınav sonuçları');`

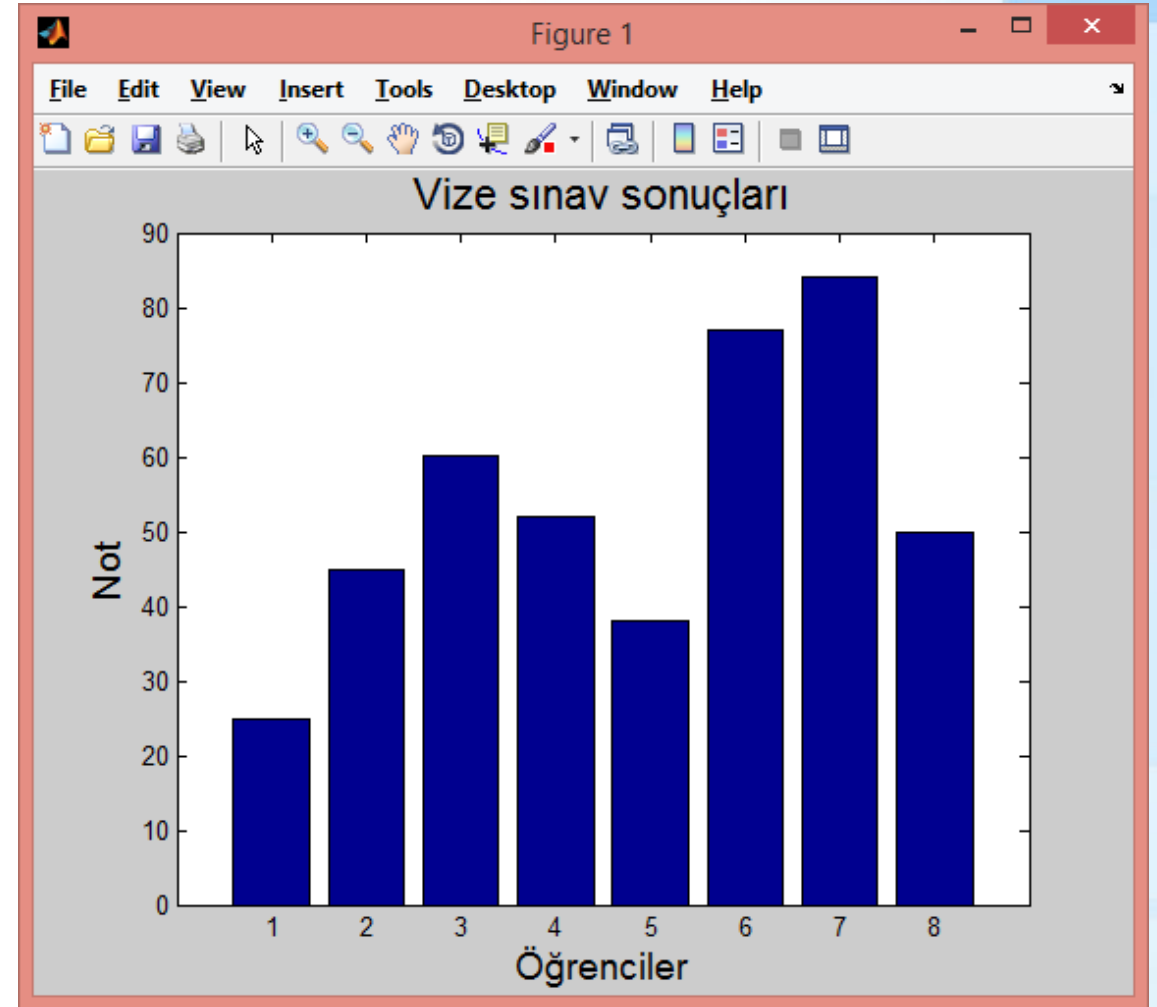
Bar Grafiği

```
>> x=[1:8];  
y=[25 45 60 52 38 77 84 50];  
bar(x,y)  
>> xlabel('Öğrenciler');  
ylabel('Not');  
title('Vize sınav sonuçları');
```



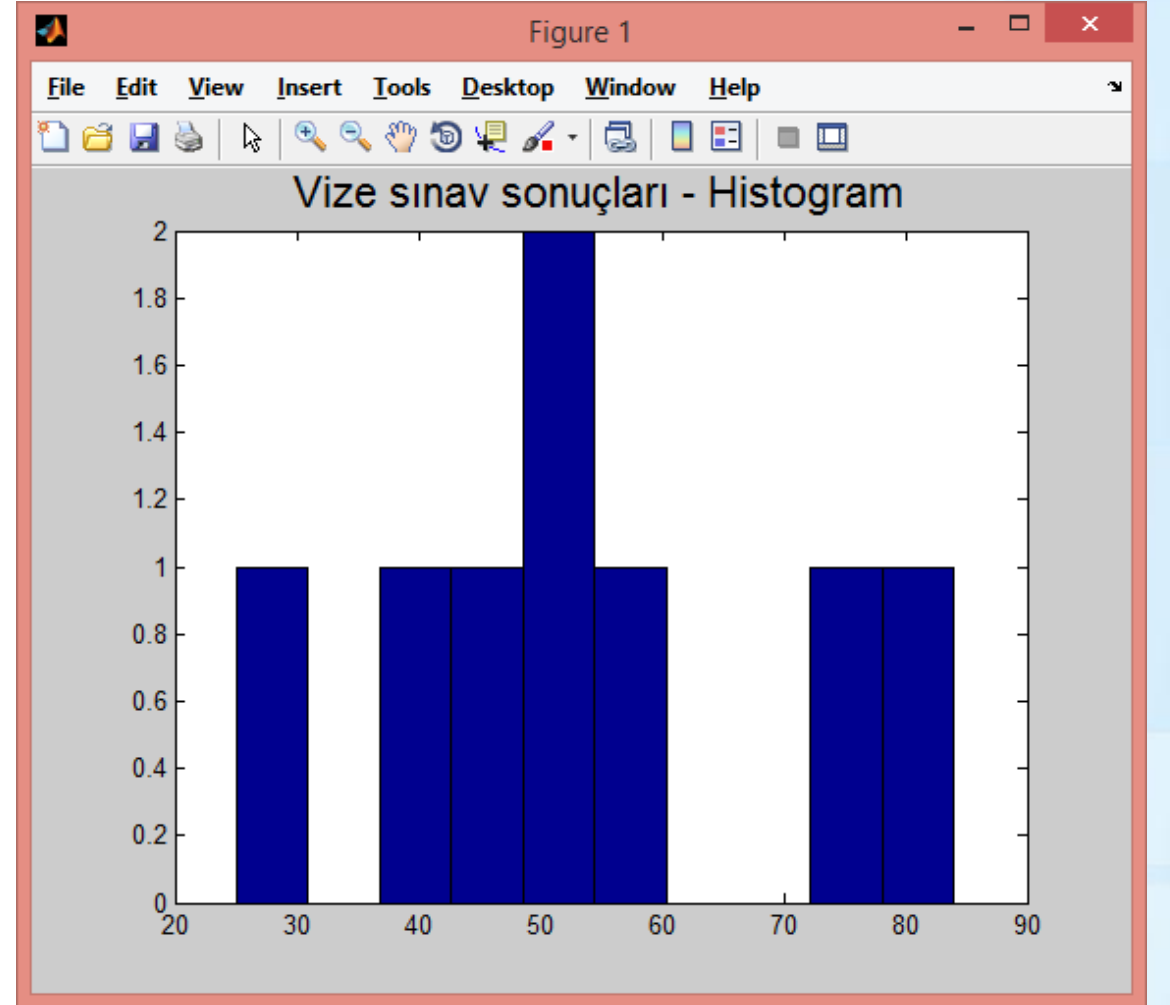
Grafik ve Eksen Başlıklarını Boyutlandırma

- `xlabel('Öğrenciler', 'fontsize', [14]);`
- `ylabel('Not', 'fontsize', 14);`
- `title('Vize sınav sonuçları', 'fontsize', 16);`



Histogram Çizimi

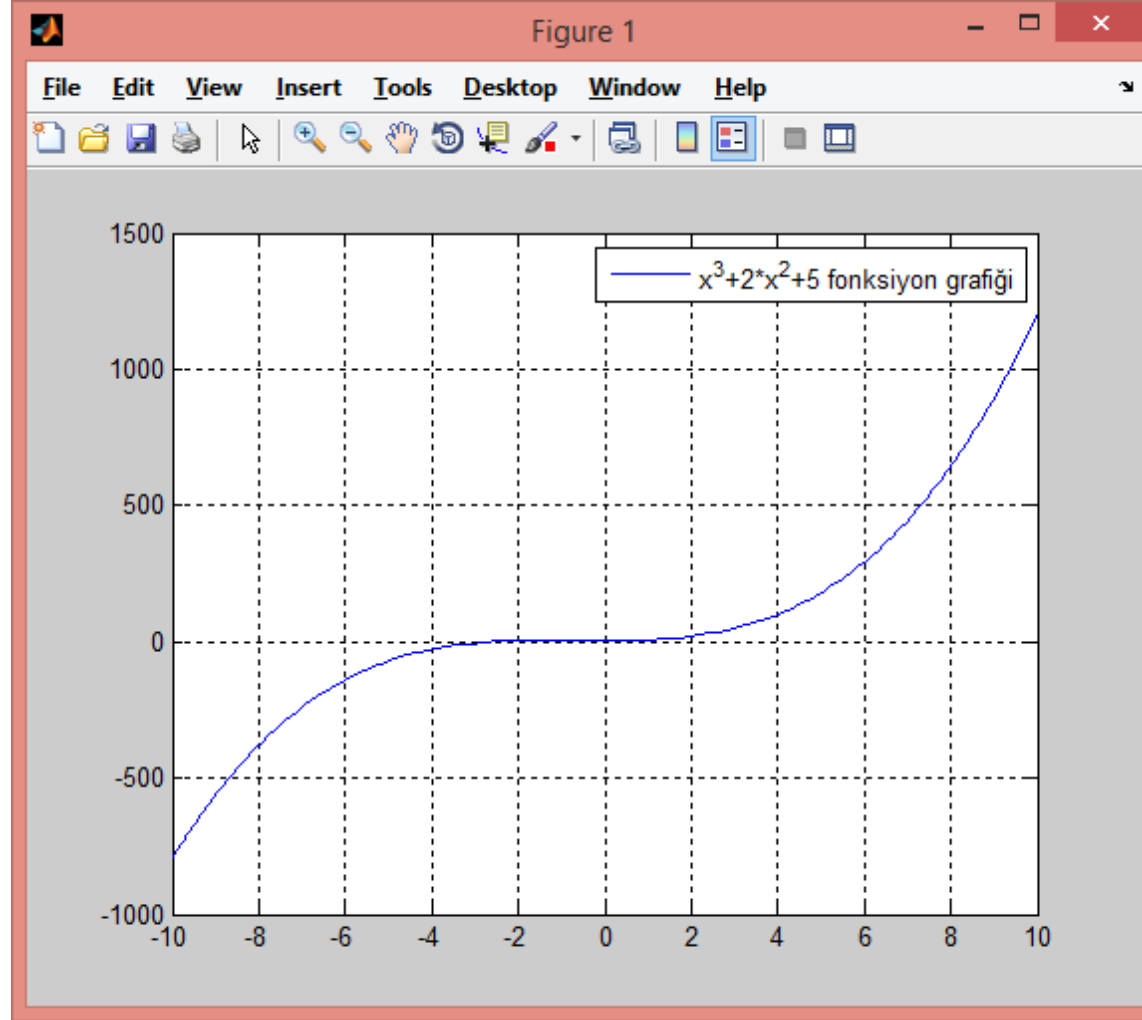
- `hist(x)` fonksiyonu ile histogram çizimi yapılır.
- 10 tane çubuk ile grafik gösterilir.
- `hist(y)`
- `title('Vize sınav sonuçları - Histogram', 'fontsize', 16);`



Fonksiyon Grafiği Çizdirmek

- `fplot ('F' , [xmin xmax])` komutu ile fonksiyon grafiği çizdirilebilir. Fonksiyon F ile belirtilen yerde tanımlanır.
- Örneğin $F(x)=x^3+2x^2+5=0$ fonksiyonunun grafiğini $[-10\ 10]$ aralığında çizelim.
- `fplot ('x^3+2*x^2+5', [-10 10])`
- `legend ('x^3+2*x^2+5 fonksiyon grafiği');`
- `grid on` komutu ile grafik üzerinde gridler gösterilebilir.
- `grid off` ile grid kaldırılır.

Fonksiyon Grafiđi izdirmek



Grafik Üzerinden Değer Okumak

- `ginput(n)` komutu ile n sayıda noktanın değeri grafik üzerinden okunabilir.
- Örneğin `ginput(4)` komutunu verirsek grafik üzerinde 4 noktanın üzerine tıklayıp x ve y koordinatlarını okuyabiliriz.
- `ginput(n)` fonksiyonu elinizin hassasiyetine göre sonuç verir.
- Kesin değer bulmak için `polyval` fonksiyonunu kullanınız.

```
>> ginput(4)
ans =
    3.9401    85.5263
   -7.2581  -265.3509
   -2.9263   -2.1930
    4.2166   129.3860
```

Polinomu Belli Bir Değer İçin Çözmek

- $F(x) = ax^4 + bx^3 + cx^2 + dx + e = 0$ şeklindeki N.inci dereceden polinomu Matlab'da $F=[a \ b \ c \ d \ e]$ matrisi olarak tanımlamak mümkündür (matris boyutu: N+1).
- Polinomun belli bir ' x_0 ' noktasında veya daha çok sayıdaki noktadaki (x_1, x_2, x_3, x_4 gibi) değerini bulmak için `polyval(F, x0)` fonksiyonu kullanılır.
- `F=[a b c d e];`
- `y=polyval(F, x0)`
- **Ya da**
- `y=polyval(F, [x1, x2, x3, x4])`

Polinomun Köklerinin Bulunması

- $F(x) = ax^4 + bx^3 + cx^2 + dx + e = 0$ şeklindeki polinomu, Matlab'da $F=[a \ b \ c \ d \ e]$ olarak tanımlayalım.
- `roots(F)` gfonksiyonu $F(x)$ 'in köklerini hesaplar.
- Örnek: $x^3 - 4x^2 + x + 6 = 0$ fonksiyonunun köklerini bulunuz.

```
>> F=[1 -4 1 6];
```

```
>> y=roots(F)
```

```
y =
```

```
3.0000
```

```
2.0000
```

```
-1.0000
```



Matlab'da İstatistik Komutları (2), Eđri Uydurma, Grafikler

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Eksik Veri Setleri ile İstatistik

- Gerçek hayatta veri setleri genellikle %100 tam değildir. Eksikler olabilir.
- Bu gibi durumlarda bir önceki derste gösterilen istatistiksel komutlar Matlab'da farklı şekilde kullanılmalıdır.

```
>> veri=[3 NaN 18 24 NaN 32 NaN 65 84 105]
veri =
     3     NaN    18    24     NaN    32     NaN    65    84   105
>> mean(veri)
ans =
     NaN
>> nanmean(veri)
ans =
    47.2857
```

Eksik Veri Setleri ile İstatistik

NaN bulunmayan verilerde

- `mean(A)`
- `max(A)`
- `min(A)`
- `sum(A)`
- `median(A)`
- `std(A)`
- `var(A)`

(Varyans)

NaN bulunan verilerde

- `nanmean(A)`
- `nanmax(A)`
- `nanmin(A)`
- `nansum(A)`
- `nanmedian(A)`
- `nanstd(A)`
- `nanvar(A)`

Eksik Veri Setleri ile İstatistik

➤ Aşağıdaki fonksiyonlar hem %100 hem de eksik veri olması durumunda aynı kullanılır:

➤ `range (A)`

➤ `quantile (A, [0.25 0.50 0.75])`

➤ `skewness (A)`

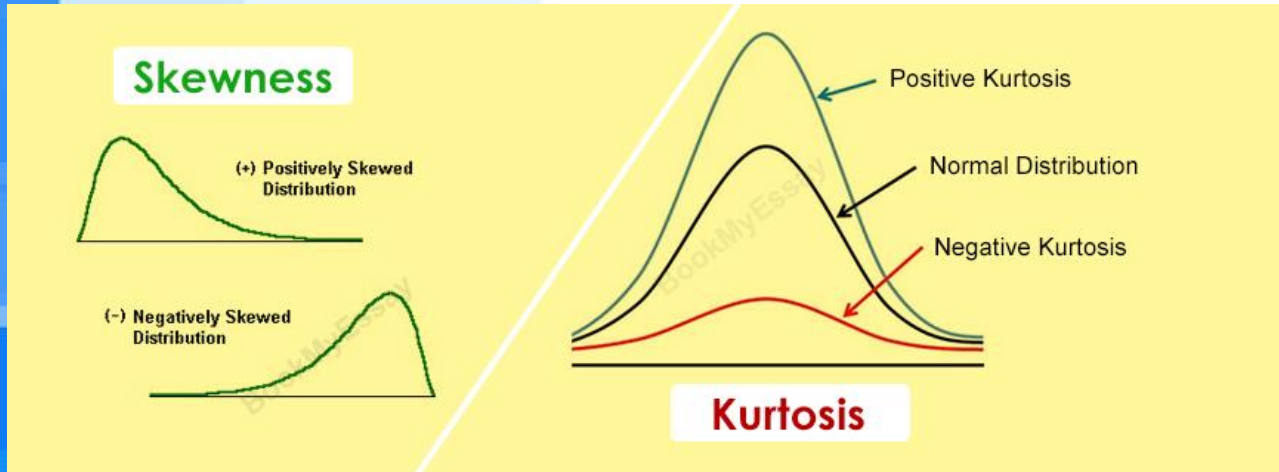
➤ `kurtosis (A)`

Aralık hesaplar (maks-min)

Çeyreklik (kantil) hesaplar.

Çarpıklık katsayısı hesaplar.

Basıklık katsayısı hesaplar.



<https://www.bookmyessay.com/skewness-and-kurtosis-assignment/>

Veri Setindeki NaN Değerler

Vektörel verideki NaN değerlerinin bulunması / değiştirilmesi

➤ NaN değerlerin sayısının bulunması:

➤ `sum(isnan(x))`

➤ NaN değerlerin temizlenerek yeni veri seti oluşturulması:

➤ `y=x(isfinite(x))`

➤ NaN değerlerin temizlenmesi:

➤ `x(isnan(x)) = []`

Veri Setindeki NaN Değerler

```
>> x=[3 NaN 18 24 NaN 32 NaN 65 84 105];  
>> sum(isnan(x))  
ans =  
      3  
>> y=x(isfinite(x))  
y =  
      3      18      24      32      65      84     105  
>> x(isnan(x))=[]  
x =  
      3      18      24      32      65      84     105
```

Doğrusal Regresyon

- En basit ifadeyle, veri setini en iyi açıklayan $y = \beta_0 + \beta_1 x$ şeklindeki doğrunun çizdirilmesidir.
- β_0 (kesim noktası) ve β_1 (eğim) regresyon katsayılarıdır.

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

Burada $\bar{y} \equiv \frac{1}{n} \sum_{i=1}^n y_i$ ve $\bar{x} \equiv \frac{1}{n} \sum_{i=1}^n x_i$ şeklindedir.

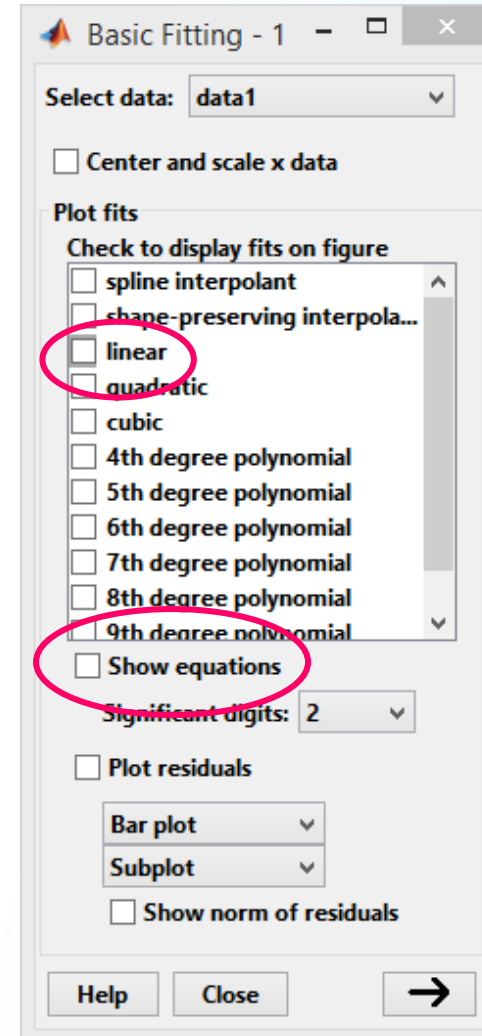
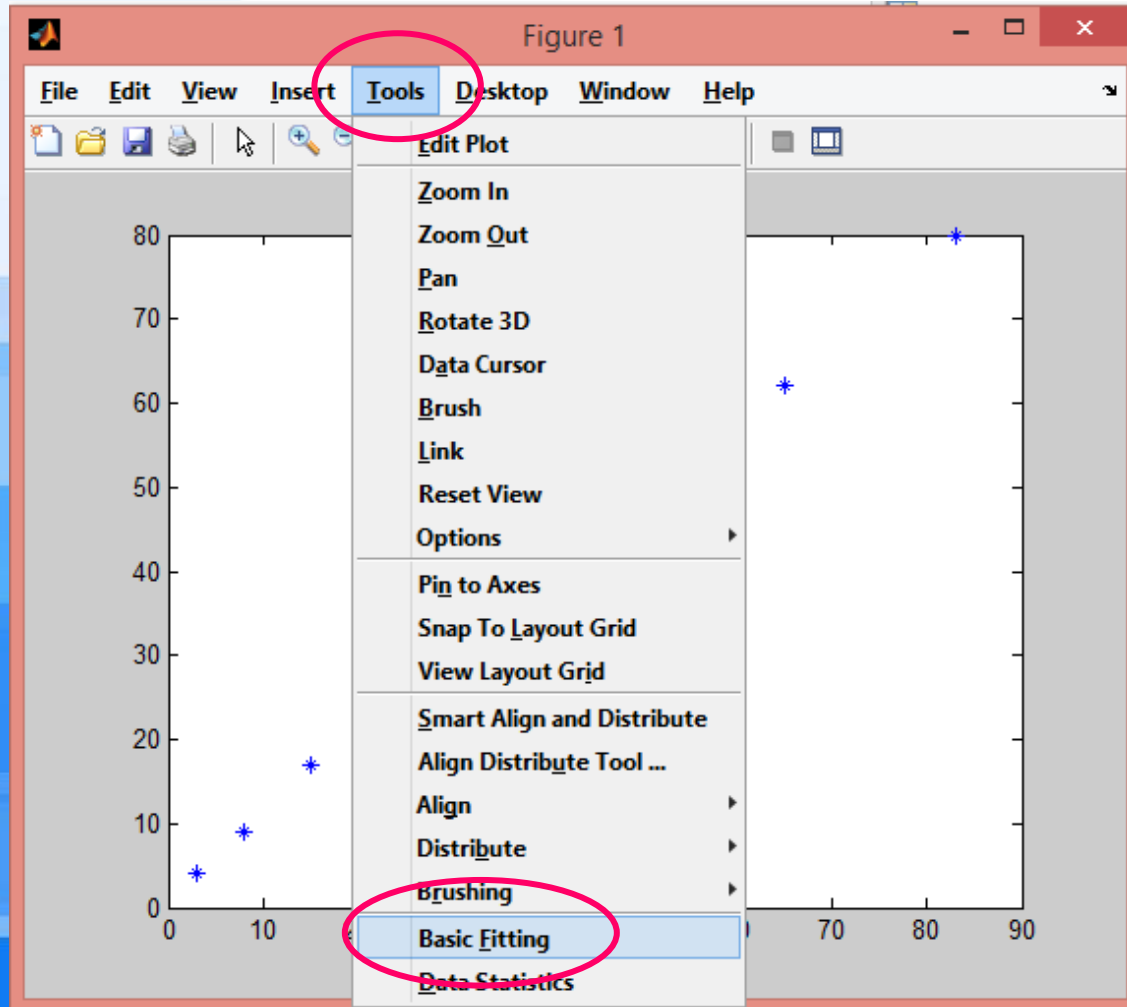
Eğri Uydurma (Curve Fitting)

- Eğri uydurulmak istenen verilerin grafiği çizilir.
- Grafik çizerken düz çizgi yerine noktalı gösterim daha uygun olacaktır.

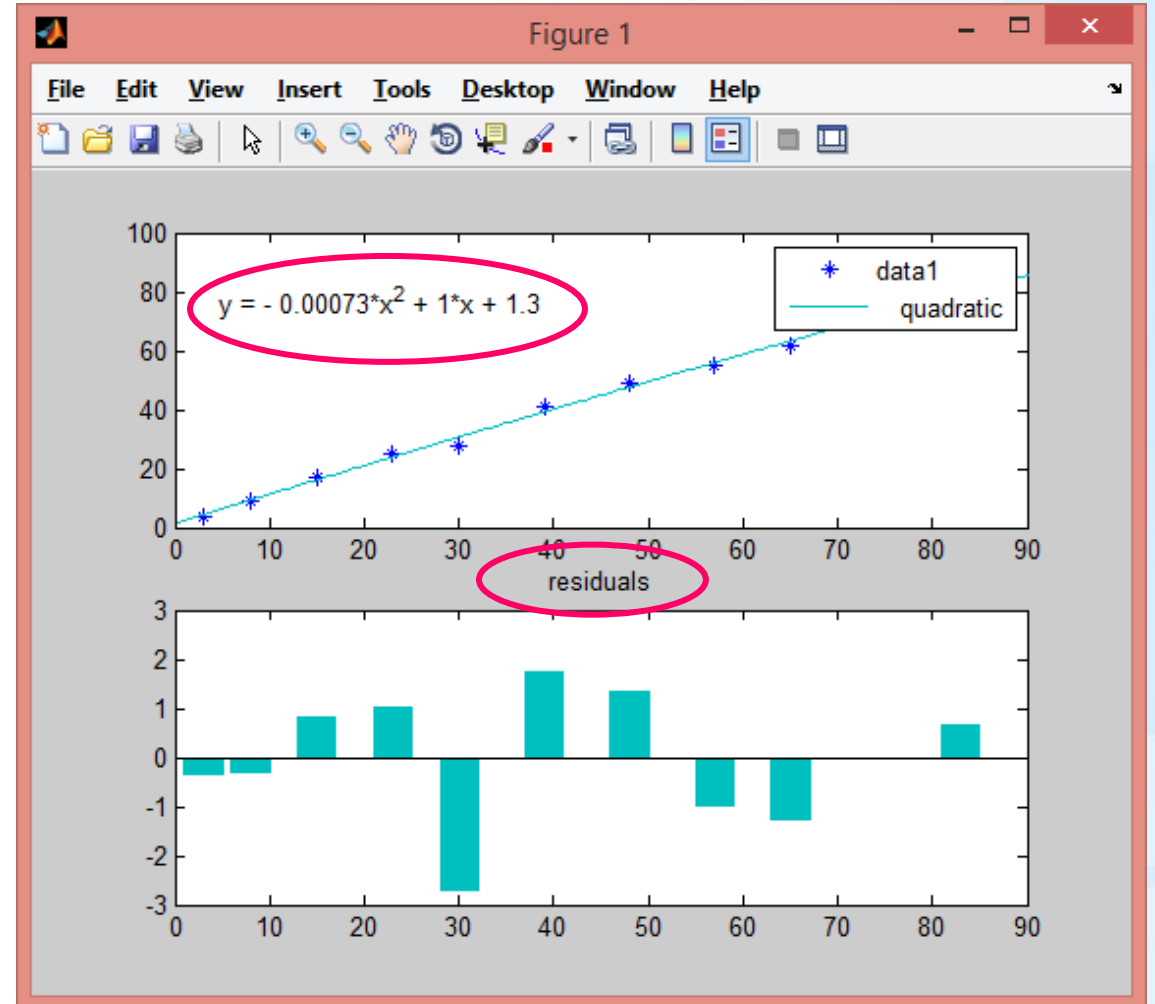
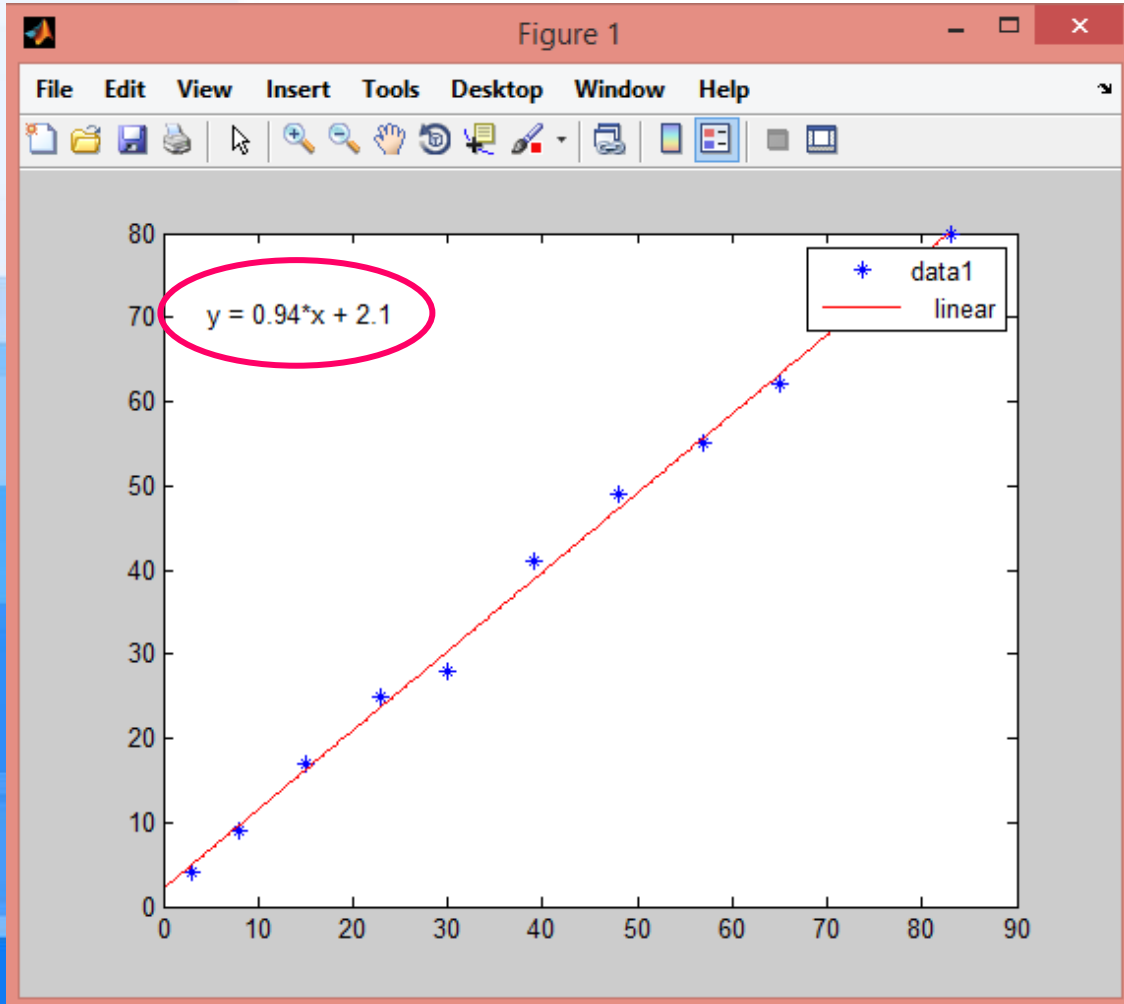
```
x=[3 8 15 23 30 39 48 57 65 83];  
y=[4 9 17 25 28 41 49 55 62 80];  
plot(x,y,'*')
```

- Grafik penceresi üzerinden **Tools > Basic Fitting** yolu izlenir.
- İstenilen eğri türü işaretlenir.
- Grafik üzerinde eğrinin denklemi görüntülenir.
- Gerekirse artıklar (residuals) grafiği de gösterilebilir.

Eğri Uydurma (Curve Fitting)



Eğri Uydurma (Curve Fitting)

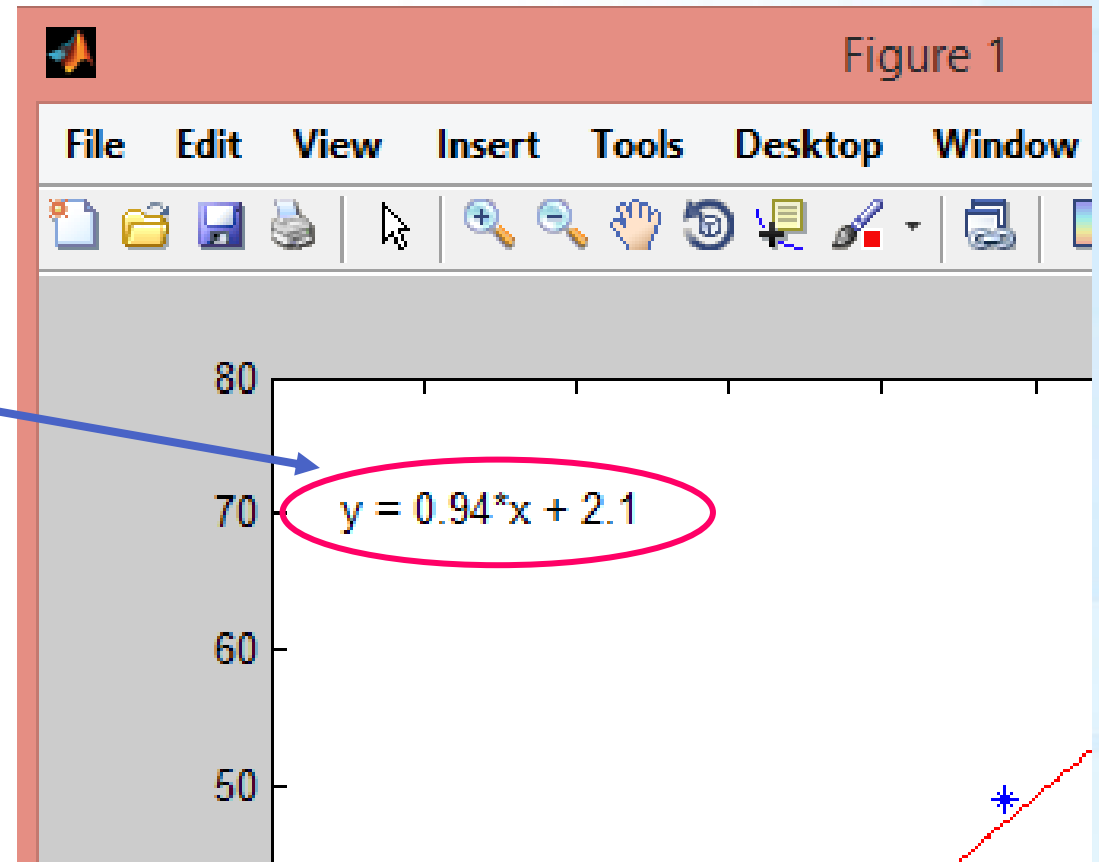


Eğri Uydurma (Curve Fitting)

- Doğru denkleminin katsayılarını elde etmek için `polyfit(x,y,n)` fonksiyonu kullanılır.

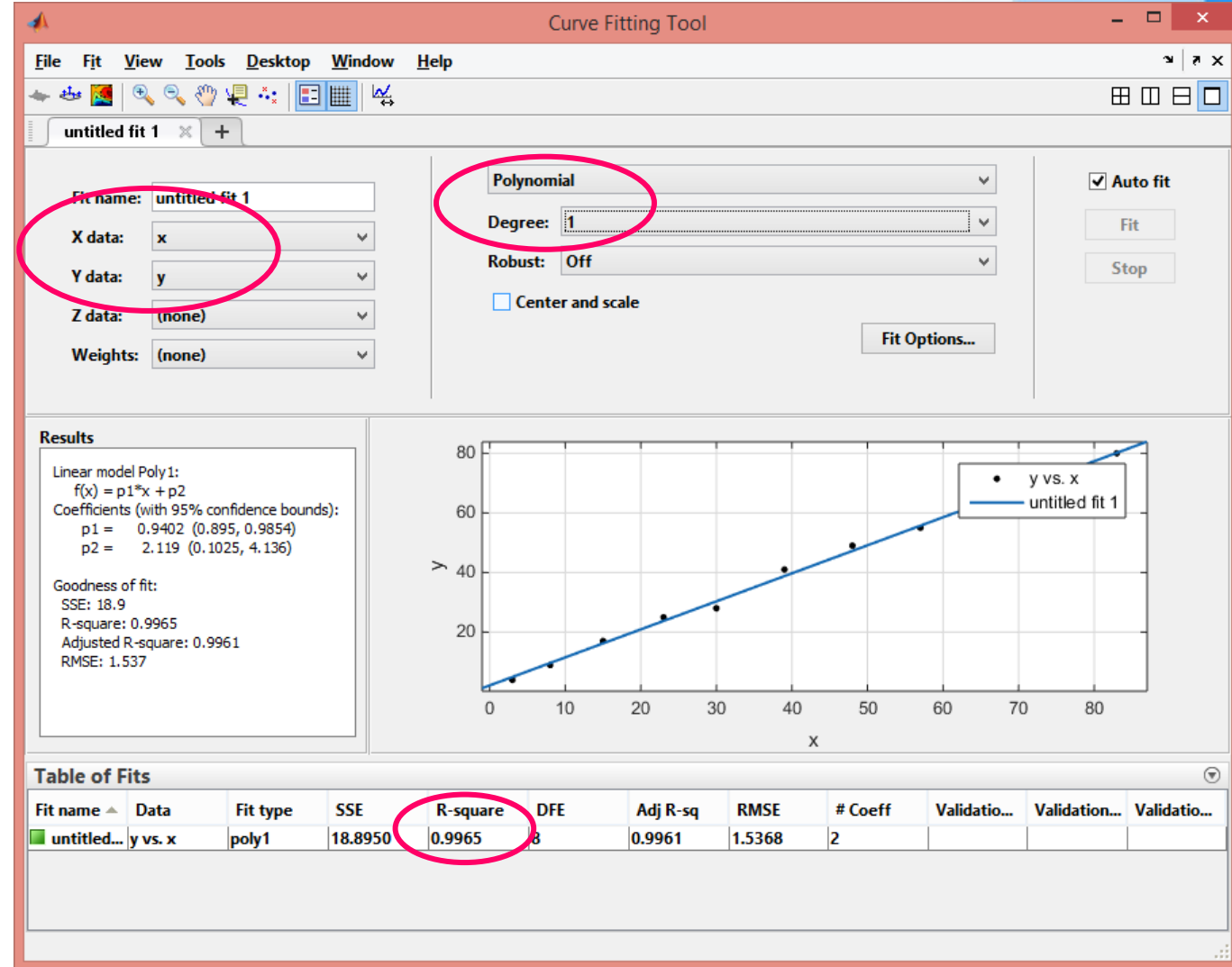
```
>> polyfit(x,y,1)  
ans =  
    0.9402    2.1190
```

```
>> polyfit(x,y,2)  
ans =  
   -0.0007    1.0002    1.3461
```



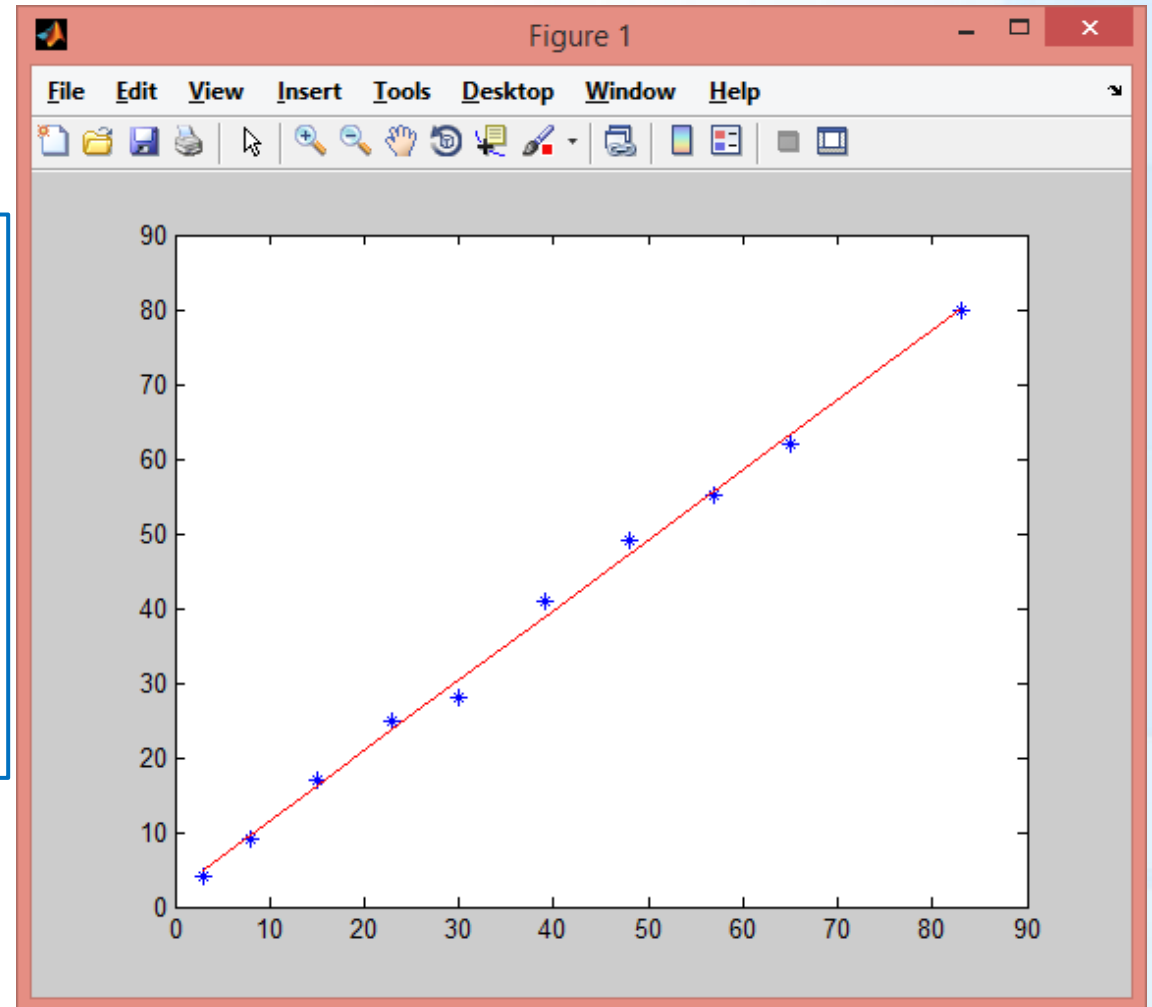
Curve Fitting Araç Kutusu

- **cftool** komutu ile **Curve Fitting** araç kutusu açılır.
- X data ve Y data seçilir.
- İstenilen fonksiyon türü seçilir.
- Uydurulan fonksiyona ait R^2 değerini görmek mümkündür.



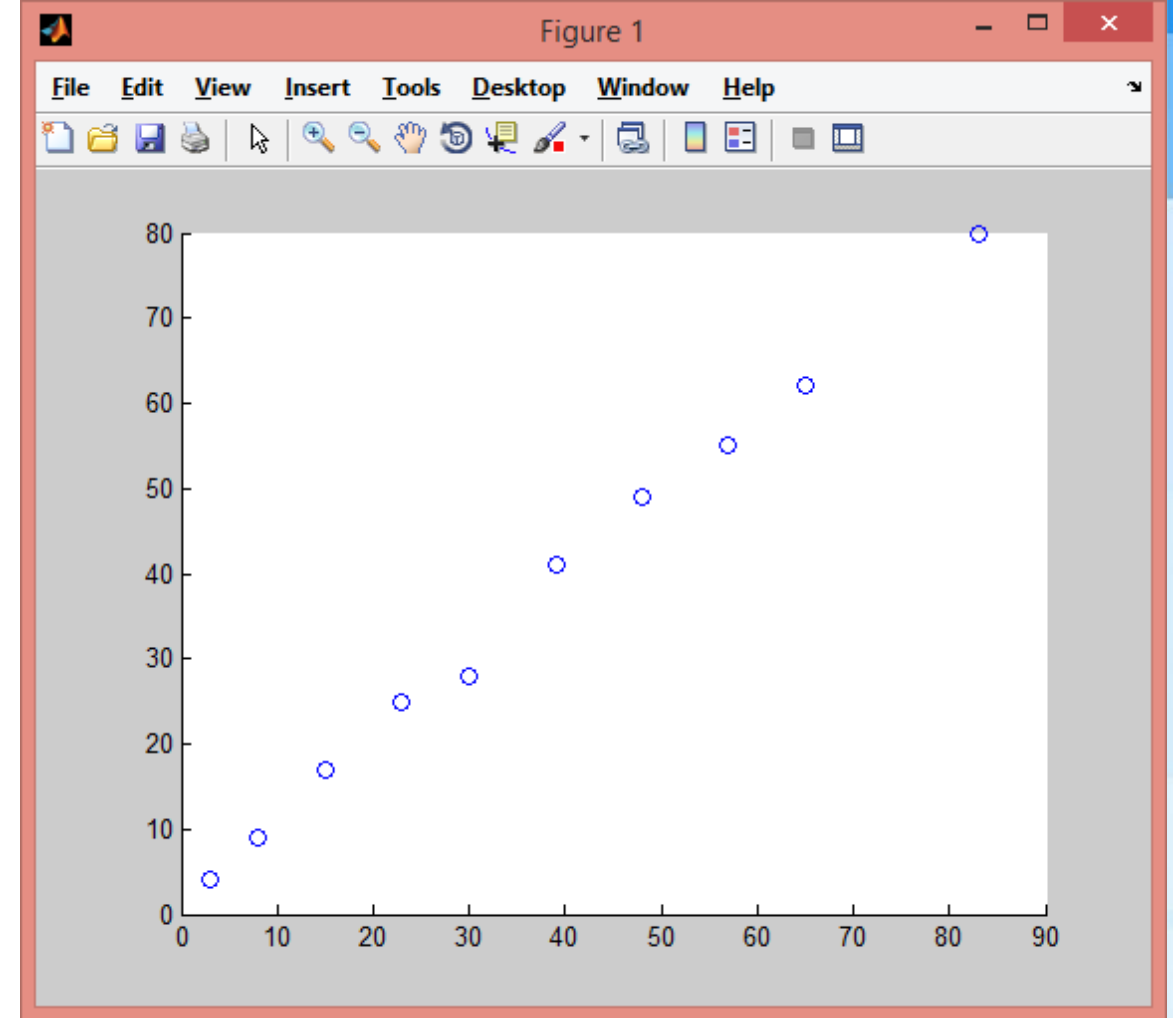
Basic Fitting Kullanmadan Eğri Uydurma

```
>> x=[3 8 15 23 30 39 48 57 65 83];  
y=[4 9 17 25 28 41 49 55 62 80];  
z=polyfit(x,y,1);  
t=polyval(z,x);  
plot(x,y,'*')  
hold on  
plot(x,t,'r')
```



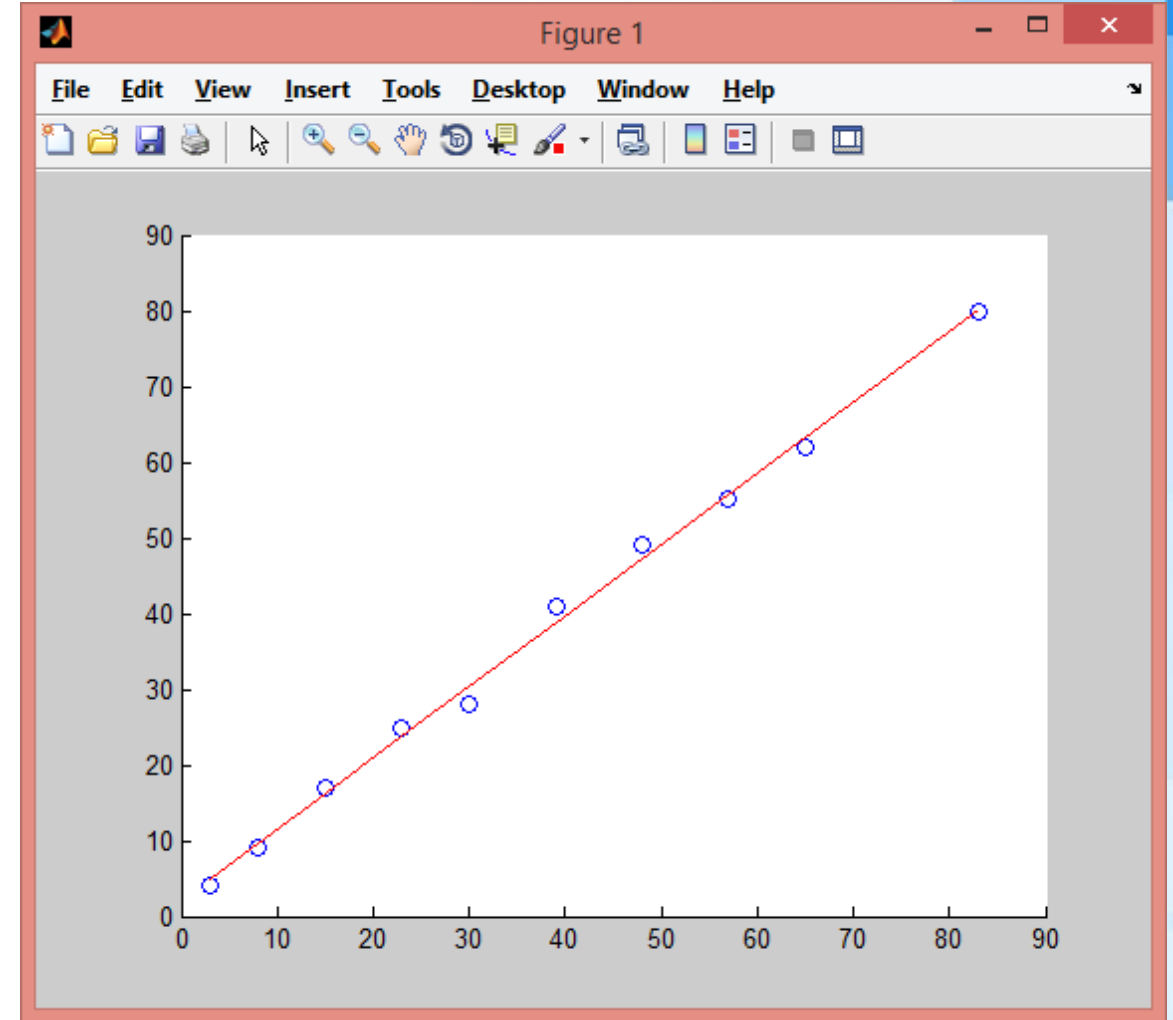
Dağılım Grafiği

- $x=[3\ 8\ 15\ 23\ 30\ 39\ 48\ 57\ 65\ 83];$
- $y=[4\ 9\ 17\ 25\ 28\ 41\ 49\ 55\ 62\ 80];$
- `scatter(x,y)` fonksiyonu ile x-y dağılım çizmek mümkündür.
- `plot(x,y,'o')` komutu ile de benzer grafik elde edilebilir.



Dağılım Grafiği Üzerinde Regresyon

- $x=[3\ 8\ 15\ 23\ 30\ 39\ 48\ 57\ 65\ 83];$
- $y=[4\ 9\ 17\ 25\ 28\ 41\ 49\ 55\ 62\ 80];$
- `scatter(x,y)`
- `hold on`
- `plot(x,polyval(polyfit(x,y,1),x),'r')`

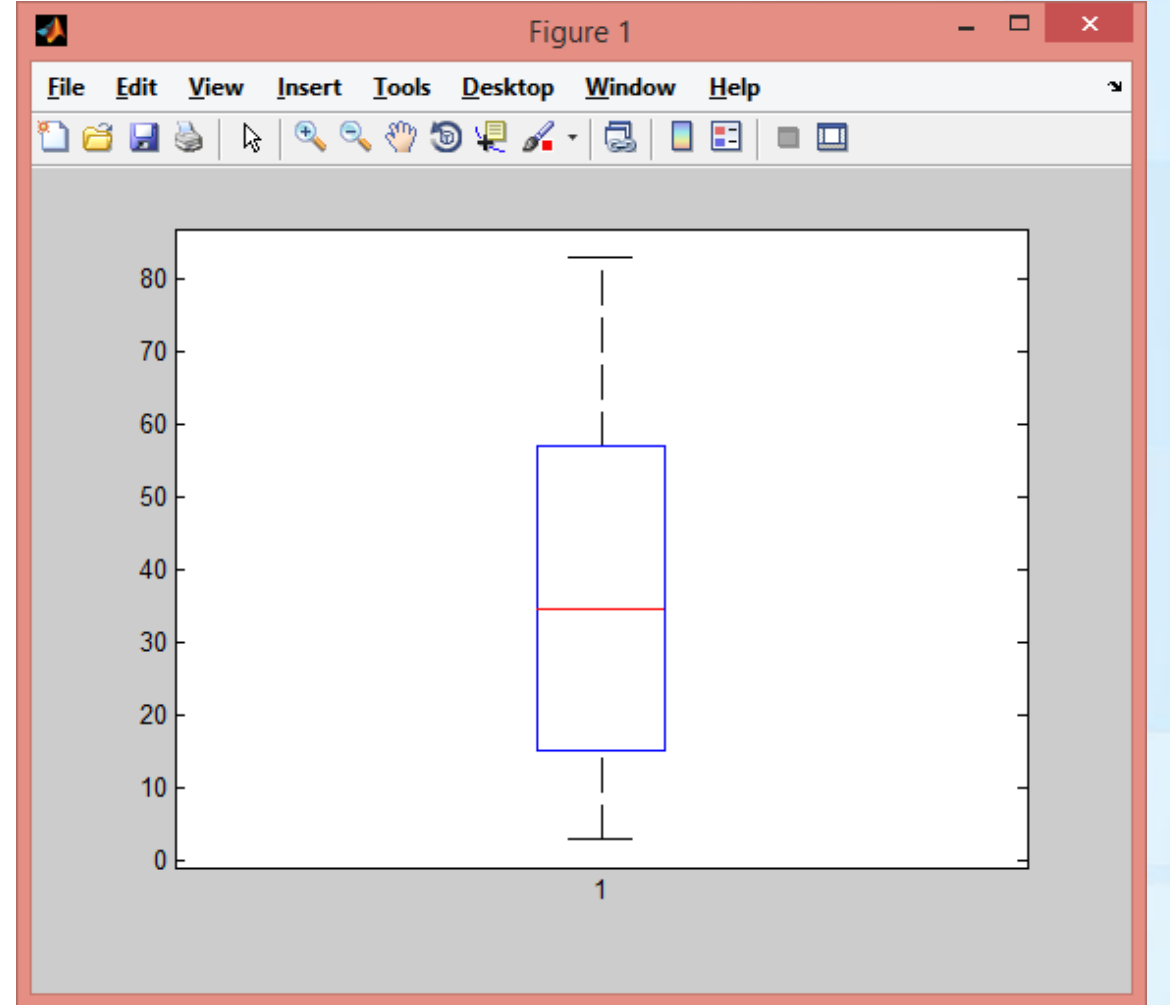


Kutu Grafiđi (Box Plot)

- **boxplot** fonksiyonu ile veri setine ait kutu grafiđi çizilir.
- $x=[3\ 8\ 15\ 23\ 30\ 39\ 48\ 57\ 65\ 83];$
- `boxplot(x)`

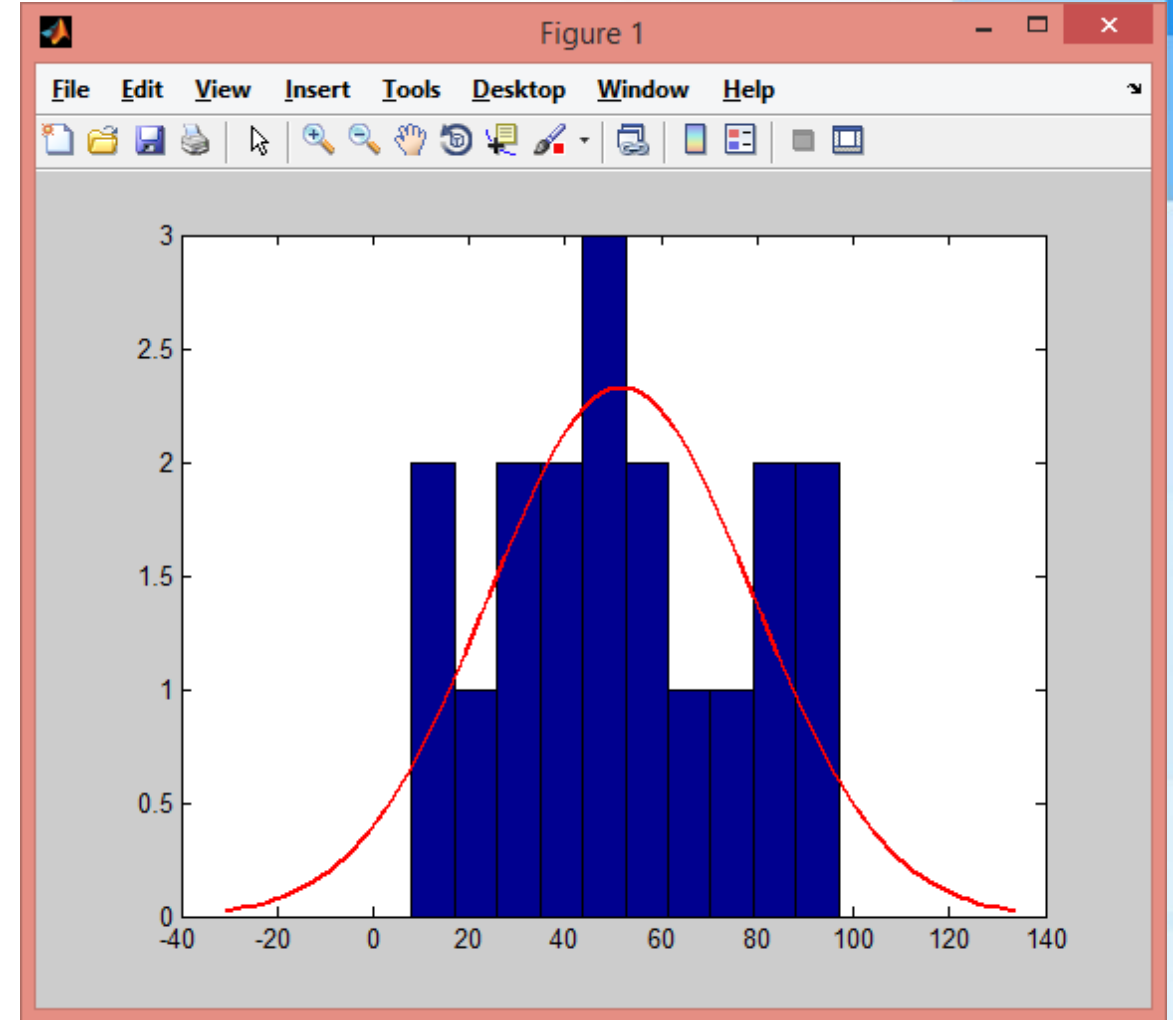


<http://www.biyoistatistikci.com/tag/kutu-grafigi/>



Histogram Üzerinde Dağılım Fonksiyonu

- **histfit** fonksiyonu ile histogram üzerinde dağılım fonksiyonu çizdirilir.
- $x=[8, 13, 17, 27, 32, 35, 42, 44, 48, 49, 54, 59, 66, 78, 83, 87, 91, 97]$;
- `histfit(x,10,'normal')`
- `histfit(veri,bin_sayısı,'dağılım_türü')`

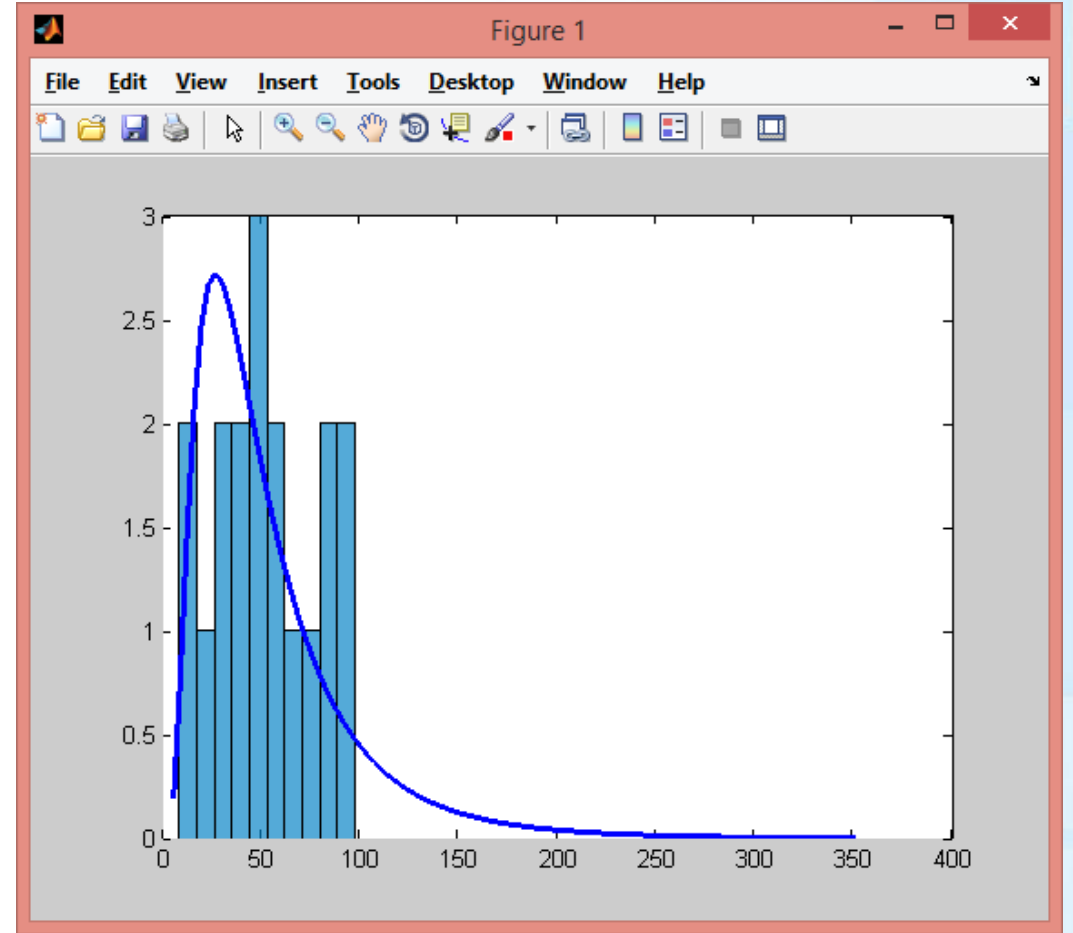



Dağılım Türleri

'beta'	Beta
'birnbaumsaunders'	Birnbaum-Saunders
'exponential'	Exponential
'extreme value' or 'ev'	Extreme value
'gamma'	Gamma
'generalized extreme value' 'gev'	Generalized extreme value
'generalized pareto' or 'gp'	Generalized Pareto (threshold 0)
'inverse gaussian'	Inverse Gaussian
'logistic'	Logistic
'loglogistic'	Log logistic
'lognormal'	Lognormal
'negative binomial' or 'nbin'	Negative binomial
'nakagami'	Nakagami
'normal'	Normal
'poisson'	Poisson
'rayleigh'	Rayleigh
'rician'	Rician
'tlocationscale'	t location-scale
'weibull' or 'wbl'	Weibull

Grafiğin Renklendirilmesi

- `histfit(x,10,'lognormal');`
- `a=histfit(x,10,'lognormal');`
- `set(a(1),'FaceColor',[0.04 0.52 0.78]);`
- `set(a(1),'FaceAlpha',[0.7]);`
- `set(a(2),'Color',[0 0 1]);`
- Renkler: [**kırmızı yeşil mavi**] 0 ile 1 arasında değer vererek ayarlanır.
- Facealpha özelliği saydamlığı ayarlar (0-1).





Matlab'da Current Folder, disp ve input fonksiyonları, Scriptler ve Fonksiyonlar

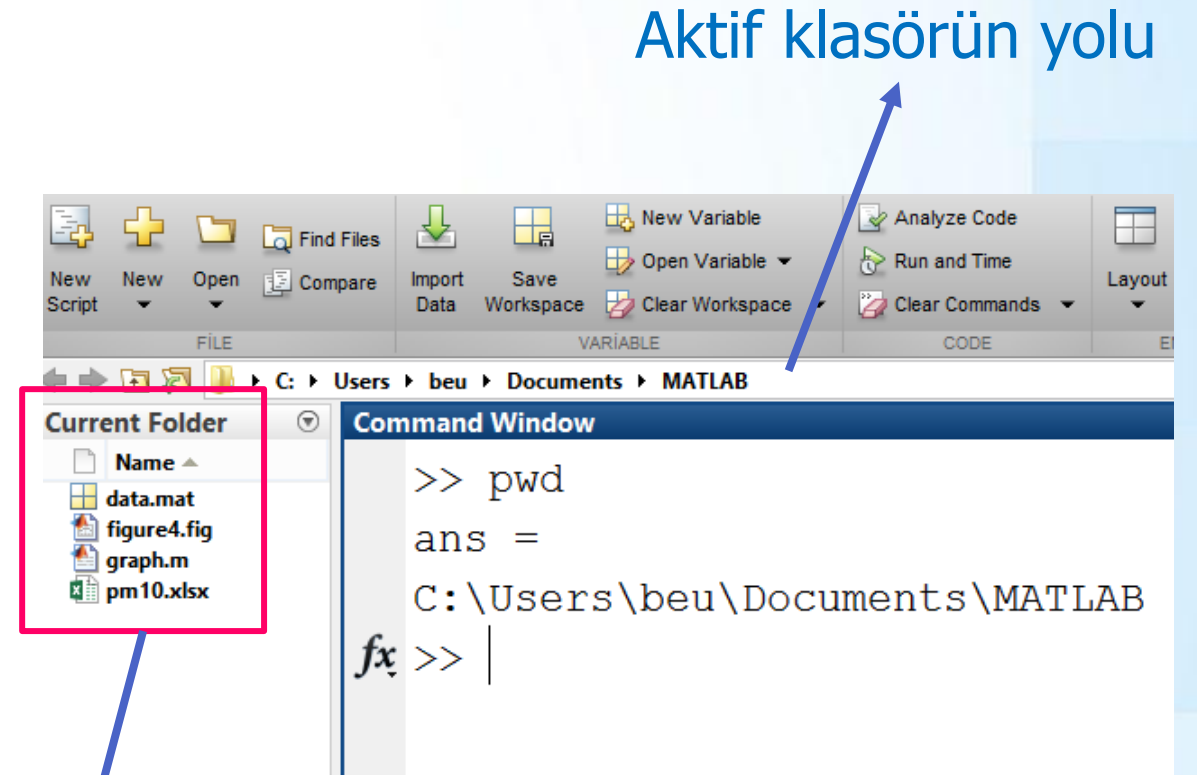
© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Current Folder (Aktif Klasör) Kavramı

- Matlab'da çalıştırılacak script ve fonksiyonlar aktif klasör içerisinde yer almalıdır. Aksi halde Matlab tarafından çalıştırılmazlar.
- Bu nedenle aktif klasör kavramının bilinmesi önemlidir.



Aktif klasör içerisinde yer alan dosyalar.

Matlab'da Klasör Komutları

- `cd,` : aktif klasörün yolunu yazdırmak için kullanılır.
- `pwd` : aktif klasörün yolunu değişkene atar.
- `mkdir dosyalar` : "dosyalar" adında yeni klasör oluşturur.
- `cd dosyalar` : "dosyalar" klasörünü aktif klasör yapar.
- `cd ..` : üst dizine çıkar.
- `what` : aktif klasörde Matlab ile ilgili dosyaları gösterir.

Matlab'da Klasör Komutları

```
>> cd,
```

```
C:\Users\beu\Documents\MATLAB
```

```
>> mkdir dosyalar
```

```
>> cd dosyalar
```

```
>> cd,
```

```
C:\Users\beu\Documents\MATLAB\dosyalar
```

```
>> cd ..
```

```
>> what
```

```
MATLAB Code files in the current folder C:\Users\beu\Documents\MATLAB
```

```
graph
```

```
MAT-files in the current folder C:\Users\beu\Documents\MATLAB
```

Matlab'da Girdi - Çıktı Komutları

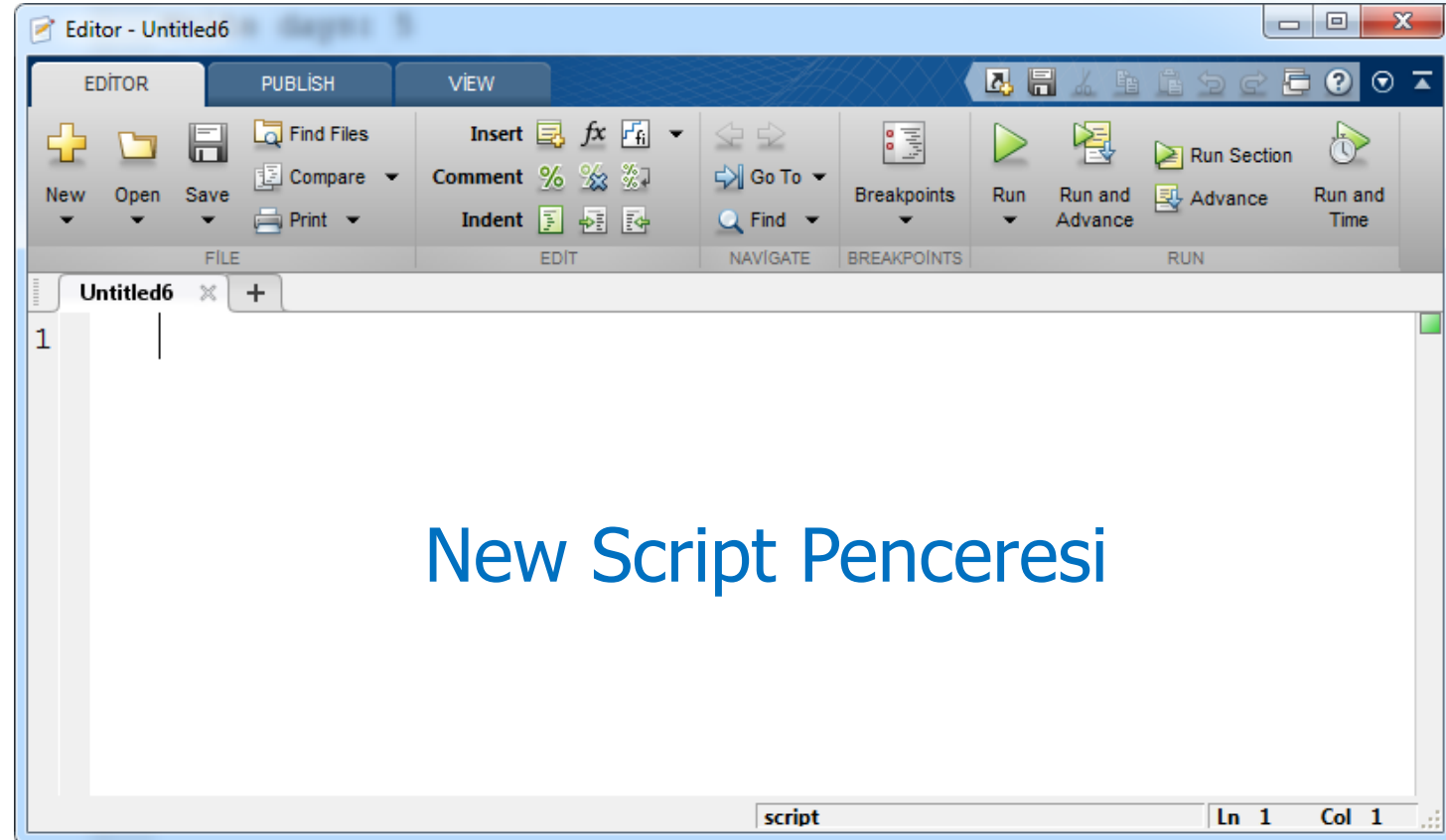
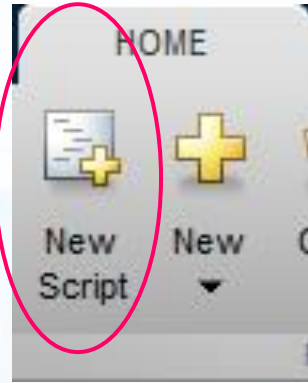
- Herhangi bir değişkenin değerinin kullanıcı tarafından girilebilmesi için `input` fonksiyonu kullanılır.
- `x=input('x değerini yazınız: ');`
- Herhangi bir değişkenin değerinin yazdırılması için `disp` fonksiyonu kullanılır.
- `disp(x)`
- `disp(['x değişkeninin değeri: ', num2str(x)])`

Matlab'da Girdi - Çıktı Komutları

```
>> x=input('x değerini yazınız: ');  
x değerini yazınız: 5  
>> disp(x)  
5  
>> disp(['x değişkeninin değeri: ',num2str(x)])  
x değişkeninin değeri: 5
```


Script Oluřturma

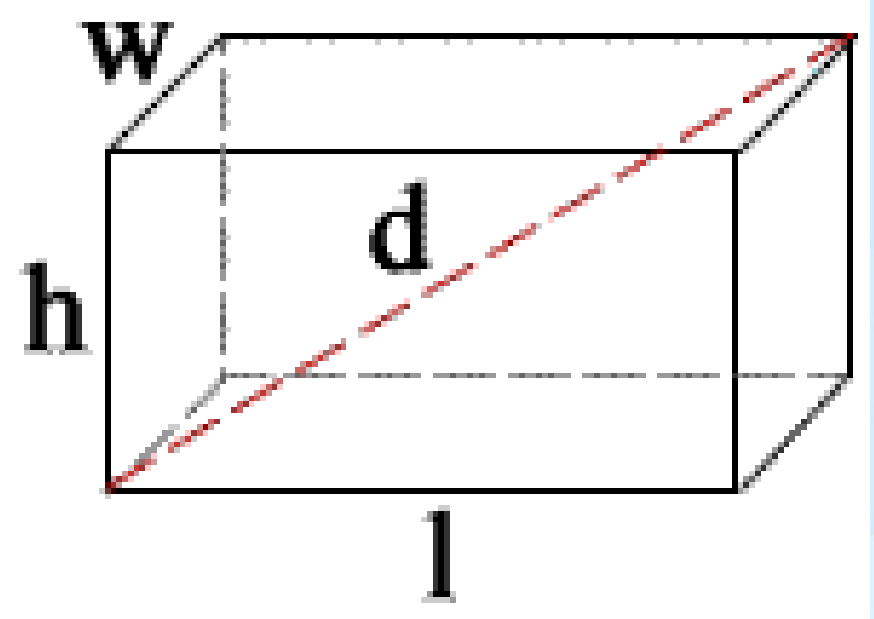
- **New Script** düğmesine tıklanır veya **CTRL + N** kısayolu kullanılır.



New Script Penceresi

Script Örneđi

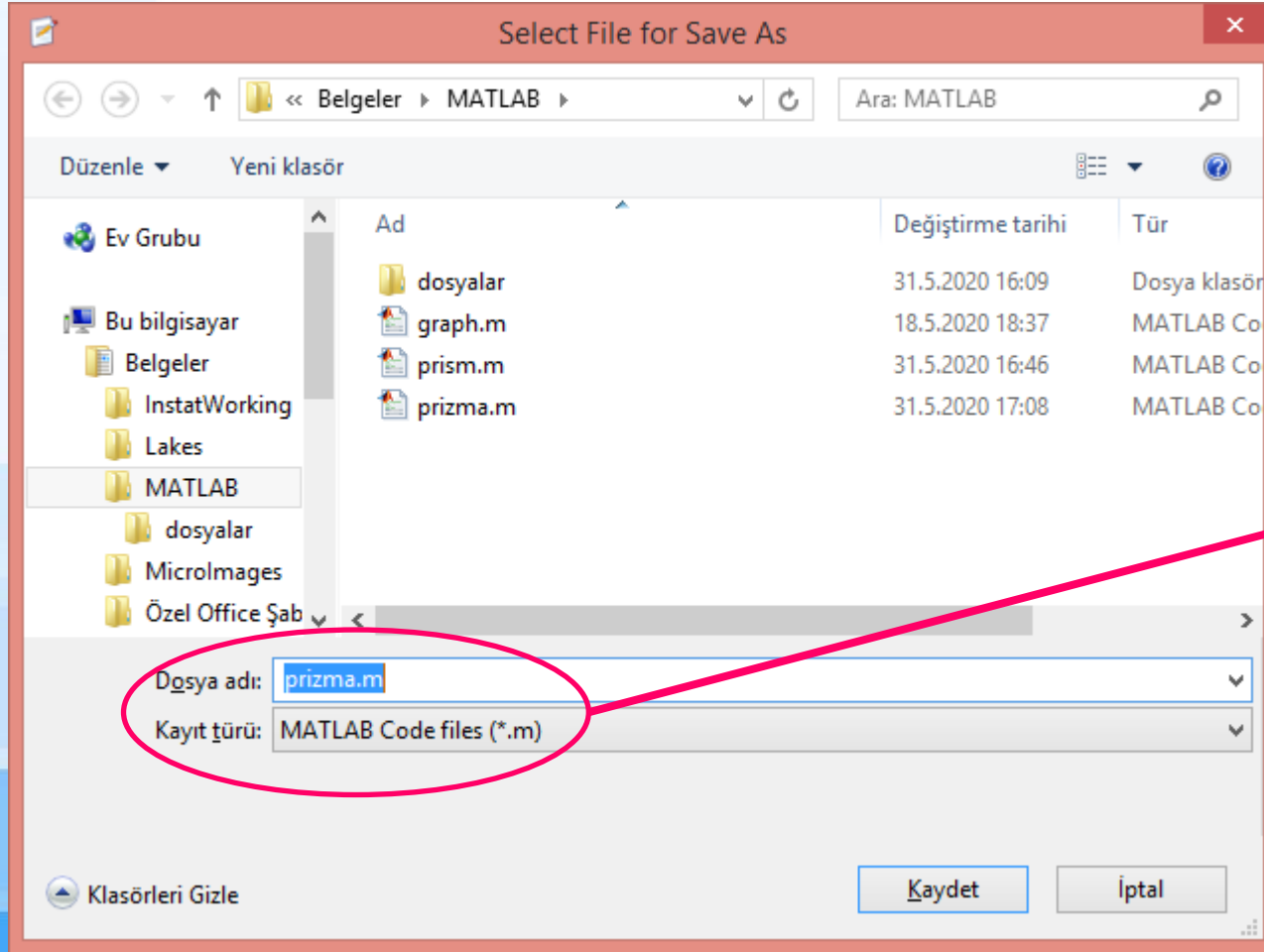
- MATLAB'da bir dikdörtgen prizmasının cisim köşegenini (d) ve hacmini hesaplayan script yazalım.
- Script öncelikle kenar uzunluklarını (l,h,w) kullanıcıdan istemelidir.
- Cisim köşegeninin (d) uzunluđu hesaplanır ve ekrana yazdırılır.
 - $d^2 = l^2 + h^2 + w^2$
- Hacim (V) hesaplanır ve ekrana yazdırılır.
 - $V = l * h * w$



Cevap

- `disp('Bu script dikdörtgen prizmasının');`
- `disp('cisim köşegenini ve hacmini hesaplar.');`
- `l=input('1.kenar uzunluğunu yazınız: ');`
- `w=input('2.kenar uzunluğunu yazınız: ');`
- `h=input('3.kenar uzunluğunu yazınız: ');`
- `d=sqrt(l^2+h^2+w^2);`
- `disp(['Köşegen Uzunluğu: ', num2str(d)]);`
- `V=l*h*w;`
- `disp(['Hacim: ', num2str(V)]);`

Scriptin Kaydedilmesi



Script adı yazılır.
Dosya *.m uzantısı
ile kaydedilir.

Script'in çalıştırılması

- Çalıştırılacak script **Aktif Klasör** içinde yer almalıdır.
- Scripti çalıştırmak için **adı yazıp Enter'a basılır** veya aktif klasörden script seçilip **F9** tuşuna basılır.

```
>> prizma
Bu script dikdörtgen prizmasının
cisim köşegenini ve hacmini hesaplar.
1.kenar uzunluğunu yazınız: 3
2.kenar uzunluğunu yazınız: 6
3.kenar uzunluğunu yazınız: 8
Köşegen Uzunluğu: 10.4403
Hacim: 144
```

Script Oluřturma - rnek Soru

- $C_xH_yO_z$ Őeklinde forml verilen bir hidrokarbonun molekl ađırlıđını hesaplamak iin script yazınız.
- Script kullanıccıdan sırasıyla x, y ve z deđerlerini ister.
 - $Ma = (x * 12) + (y * 1) + (z * 16);$
- Script ekrana hidrokarbonun molekl ađırlıđını yazar.

Script Oluşturma - Cevap

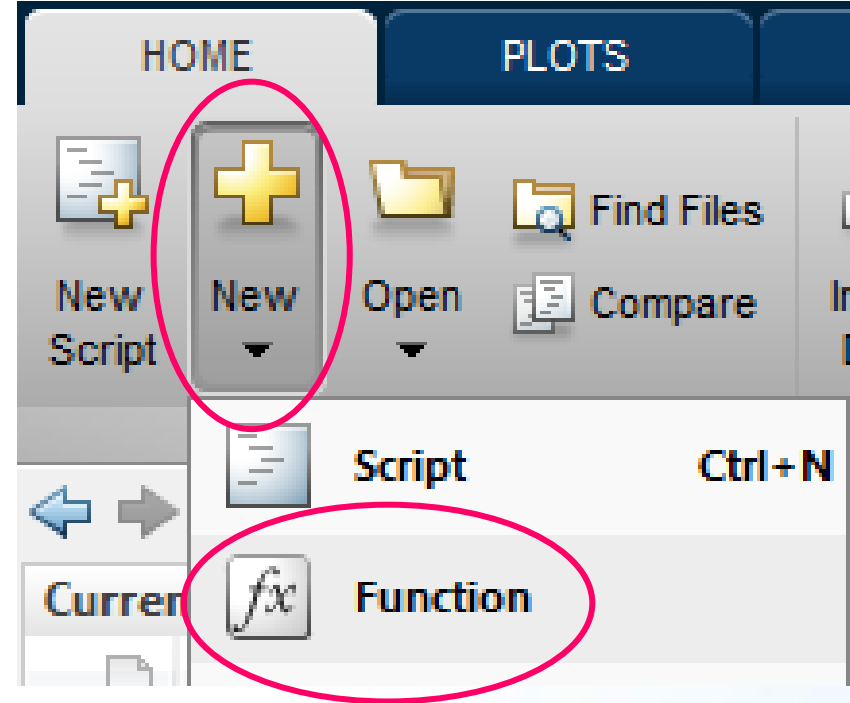
- `disp('CxHyOz hidrokarbon molekül ağırlığı hesap programı');`
- `x=input('Karbon sayısını yazınız: ');`
- `y=input('Hidrojen sayısını yazınız: ');`
- `z=input('Oksijen sayısını yazınız: ');`
- `Ma=(x*12)+(y*1)+(z*16);`
- `disp(['Hidrokarbon ağırlığı:
, num2str(Ma)]);`

Ekran Görüntüsü

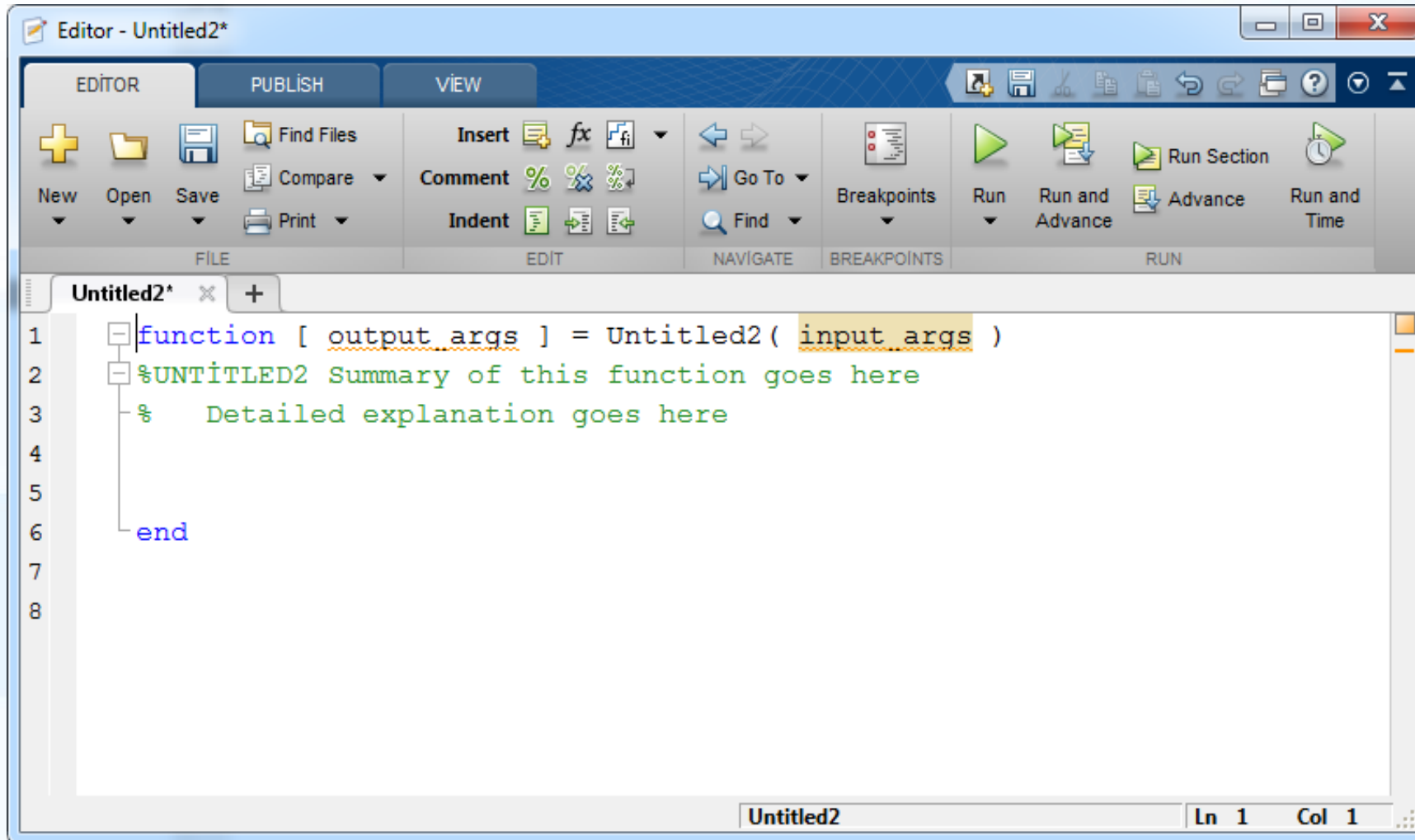
```
>> hidrokarbon  
CxHyOz hidrokarbon molekül ağırlığı hesap programı  
Karbon sayısını yazınız: 2  
Hidrojen sayısını yazınız: 5  
Oksijen sayısını yazınız: 1  
Hidrokarbon ağırlığı: 45
```


Fonksiyon Oluřturmak

- Fahrenheit deęerini Celsius deęerine evirmek iin **cel2fah** adında bir fonksiyon oluřturalım.
- Formlmz:
$$f=(c*9/5)+32$$
- **New > Function** yolu izlenerek editr penceresi aılır.



Editor Penceresi



"cel2fah" Fonksiyonunun Yapısı

```
1 function f = cel2fah(c)
3 % cel2fah function converts C to F
4 % c is input, f is output
5 |
6 f = (c*9/5) + 32;
7
8 end
```

Fonksiyonun çıktısı

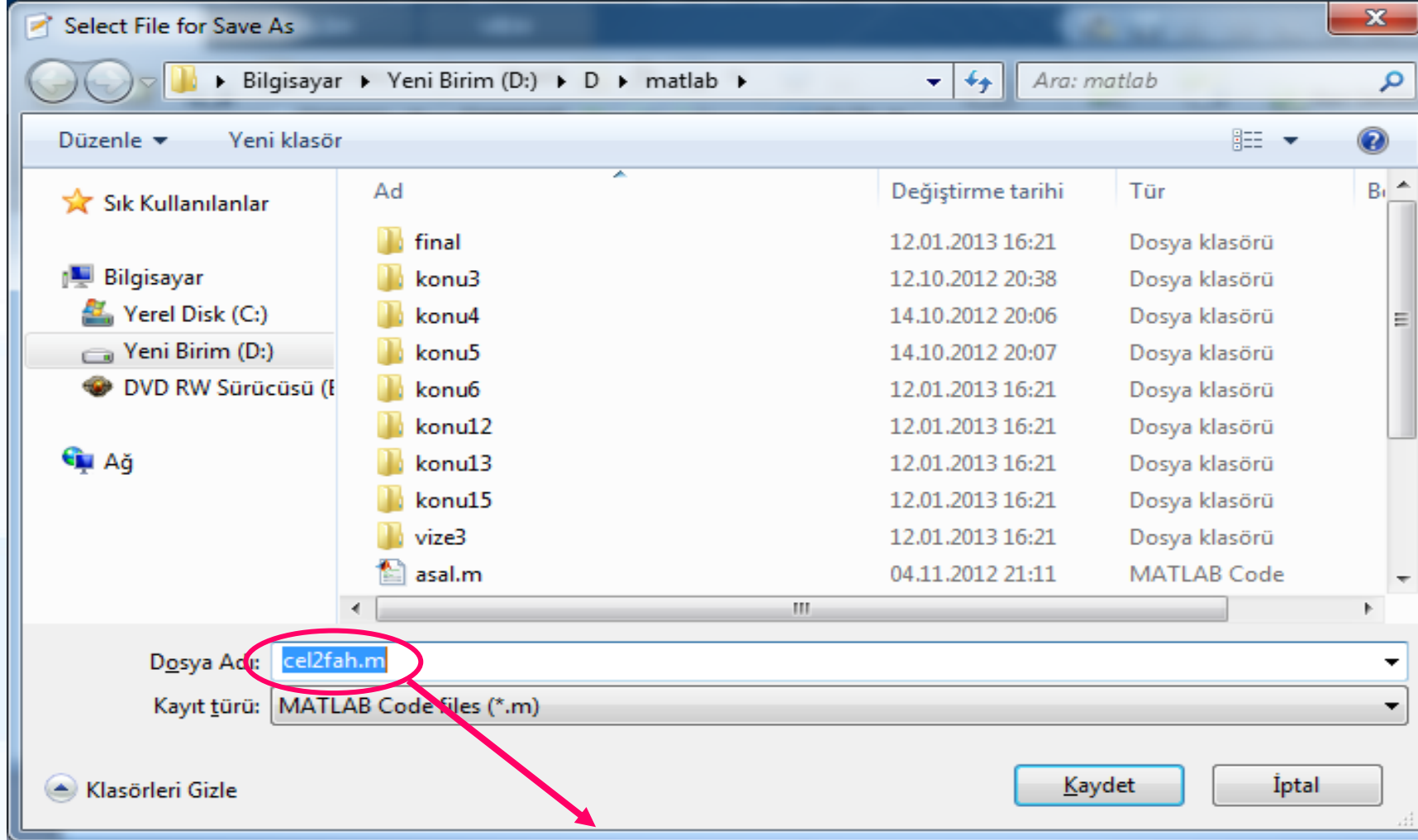
Fonksiyon adı

Fonksiyonun girdisi

Yorum satırları

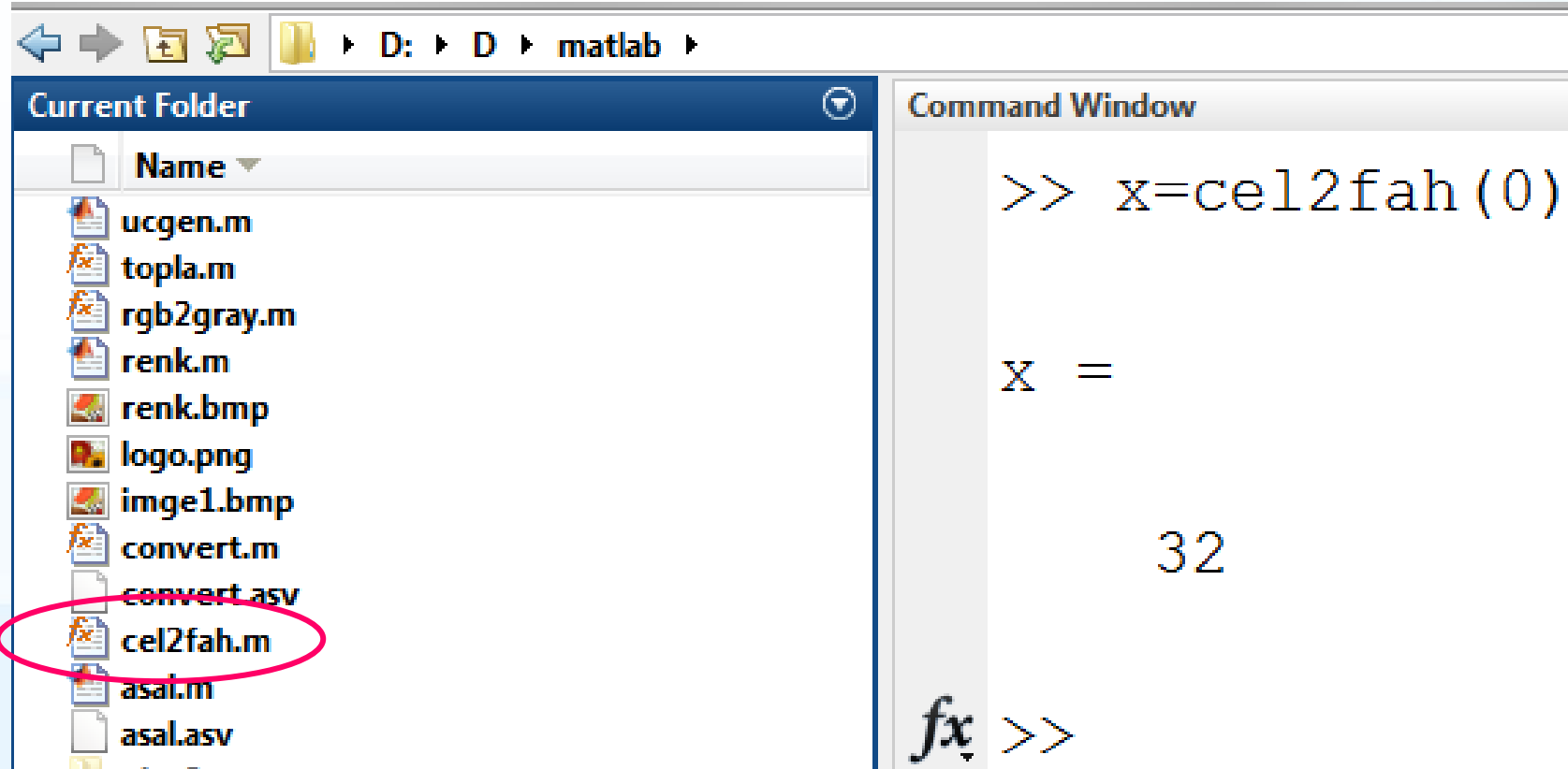
Fonksiyon Komutları

Fonksiyonun Kaydedilmesi



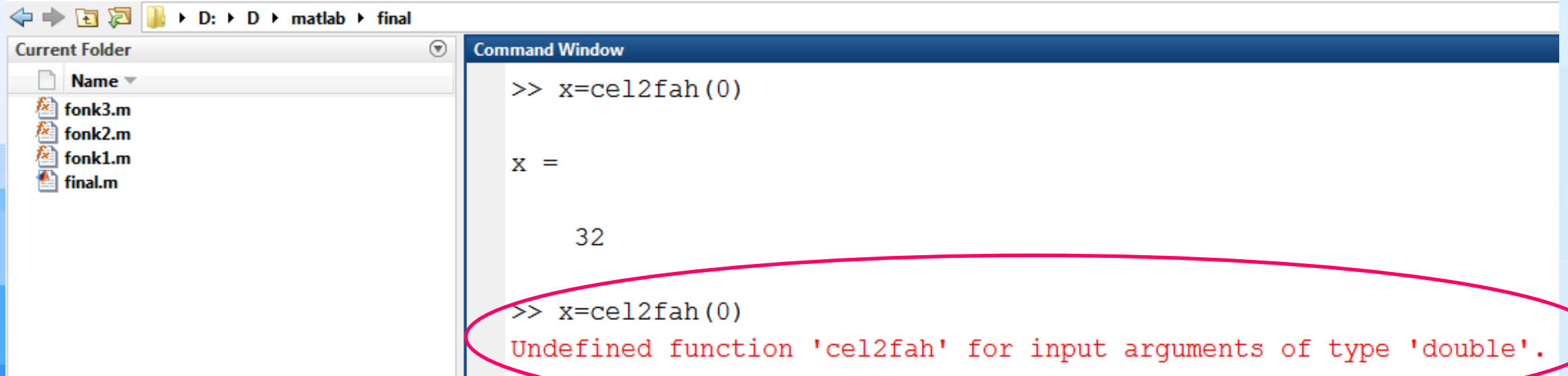
Değiştirmeyin

Yeni Fonksiyonun Kullanılması



- `cel2fah` fonksiyonu, aktif klasör içerisindeyken kullanılabilir.

Aktif Klasörde Olmayan Fonksiyonu Çağırarak



```
>> x=cel2fah(0)

x =

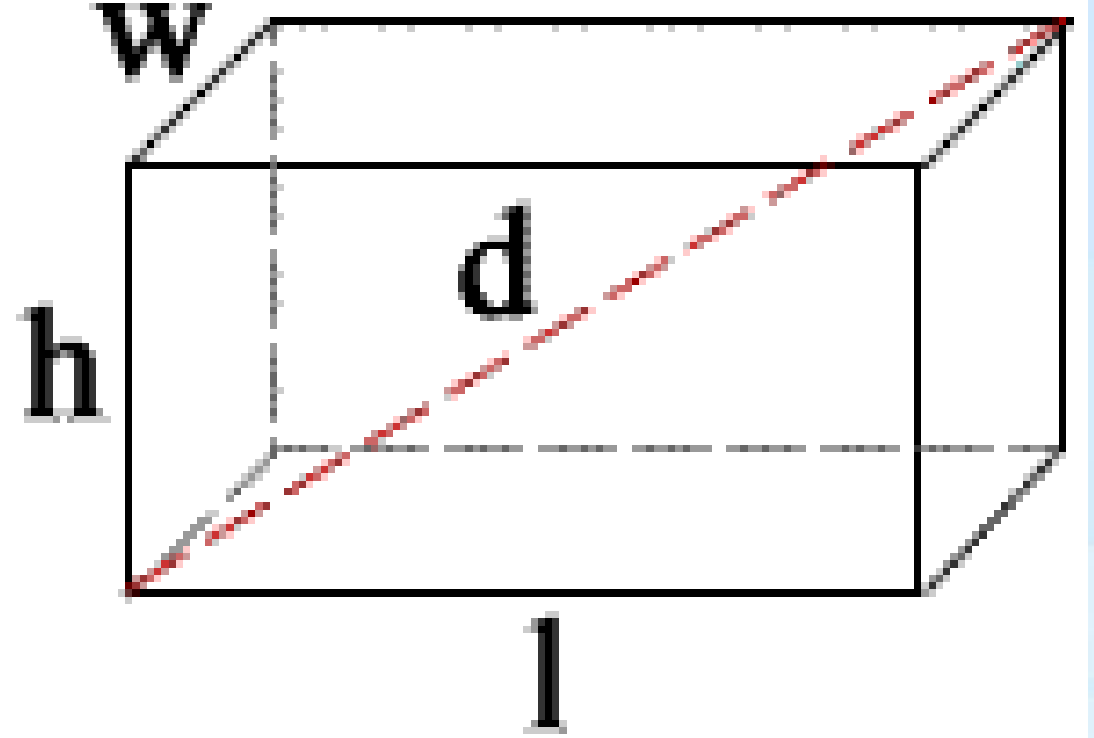
    32

>> x=cel2fah(0)
Undefined function 'cel2fah' for input arguments of type 'double'.
```

- Aktif klasör içerisinde yer almayan fonksiyon çalıştırılmak istendiğinde Matlab tanımsız fonksiyon uyarısı verecektir.

Fonksiyon Örneđi

- MATLAB'da uzunluk ölçüleri verilen dikdörtgen prizmasının cisim köşegenini (d) ve hacmini hesaplayan bir fonksiyon yazalım.
 - $d^2 = l^2 + h^2 + w^2$
 - $V = l * h * w$
- Fonksiyon sonuçları ekrana yazdırsın,
- Fakat geriye bir deęer döndürmesin.



Cevap

- `function prism(l,h,w)`
- `%prism fonksiyonu hakkında tanımlar...`
- `d=sqrt(l^2+h^2+w^2);`
- `disp(['Köşegen Uzunluğu: ',num2str(d)])`
- `V=l*h*w;`
- `disp(['Hacim: ',num2str(V)])`
- `end`

Fonksiyonun Ekran Çıktısı

- Fonksiyon adı üç girdi değeri ile birlikte yazılırsa fonksiyon çalışacaktır.
- Fonksiyon hakkında yardım almak için `help prism` yazılır.

```
>> prism(3,6,8)
```

```
Köşegen Uzunluğu: 10.4403
```

```
Hacim: 144
```

```
>> help prism
```

```
prism fonksiyonu hakkında tanımlar...
```

Fonksiyon Hakkında Açıklama Oluşturulması

- `function prism(l,h,w)`
- `%prism` fonksiyonu dikdörtgen prizmanın cisim köşegenini (d) ve hacmini (V) hesaplar.
- `%Sonuçları Ekrana yazdırır.`
- `%Fonksiyonun girdileri prizmanın kenar uzunluklarıdır.`
- `d=sqrt(l^2+h^2+w^2);`
- `disp(['Köşegen Uzunluğu: ',num2str(d)])`
- `V=l*h*w;`
- `disp(['Hacim: ',num2str(V)])`
- `end`

Fonksiyon Yardımı

```
>> help prism
```

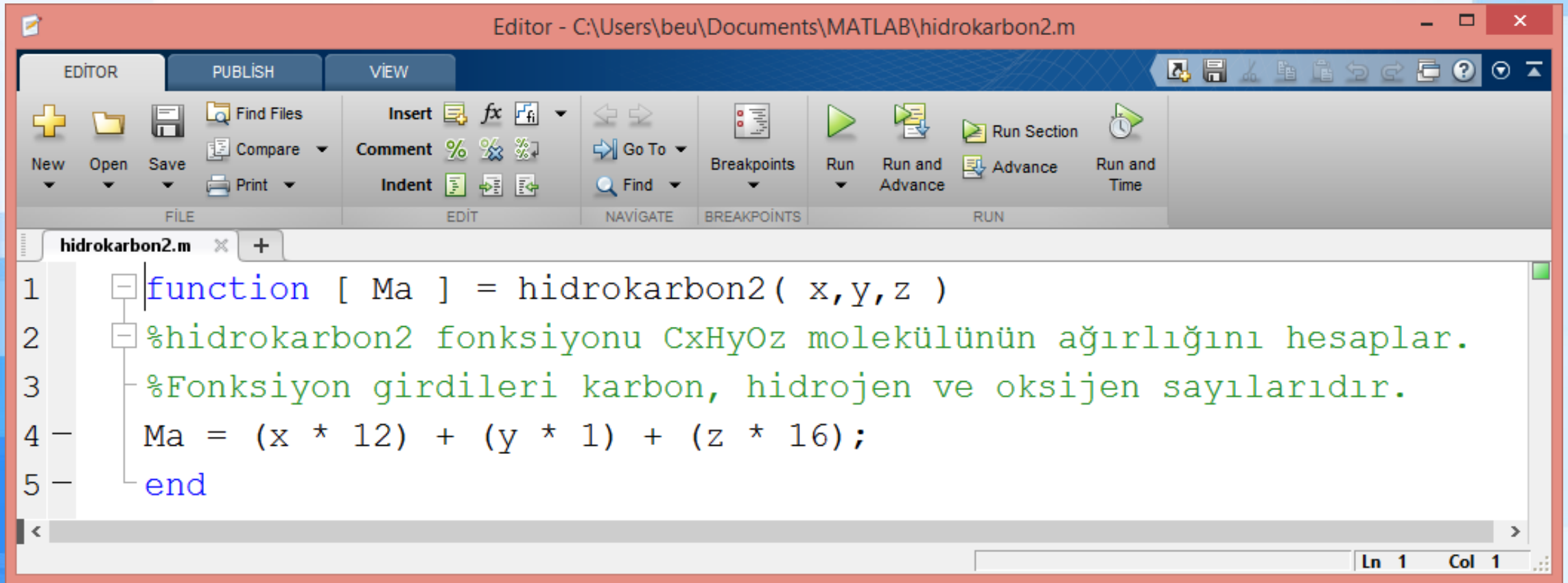
```
prism fonksiyonu dikdörtgen prizmanın cisim köşegenini (d) ve hacmini  
Sonuçları Ekranaya yazdırır.
```

```
Fonksiyonun girdileri prizmanın kenar uzunluklarıdır.
```

Fonksiyon Örneği

- Hidrokarbon ($C_xH_yO_z$) molekül ağırlığı hesaplamak için yazılan script fonksiyon olarak yazalım.
- Fonksiyonun girdileri sırasıyla x, y ve z.
- Fonksiyon çıktısı Ma.
 - $Ma = (x * 12) + (y * 1) + (z * 16);$

Fonksiyon Örneği



The screenshot shows the MATLAB Editor interface with the following code in the editor window:

```
1 function [ Ma ] = hidrokarbon2( x,y,z )  
2 %hidrokarbon2 fonksiyonu CxHyOz molekülünün ağırlığını hesaplar.  
3 %Fonksiyon girdileri karbon, hidrojen ve oksijen sayılarıdır.  
4 Ma = (x * 12) + (y * 1) + (z * 16);  
5 end
```

The status bar at the bottom right of the editor window indicates "Ln 1 Col 1".

```
>> hidrokarbon2(2,5,1)  
ans =  
    45
```



Matlab'da Deęişken Türleri, Dış Veri Aktarımı

© Doç. Dr. Özgür ZEYDAN

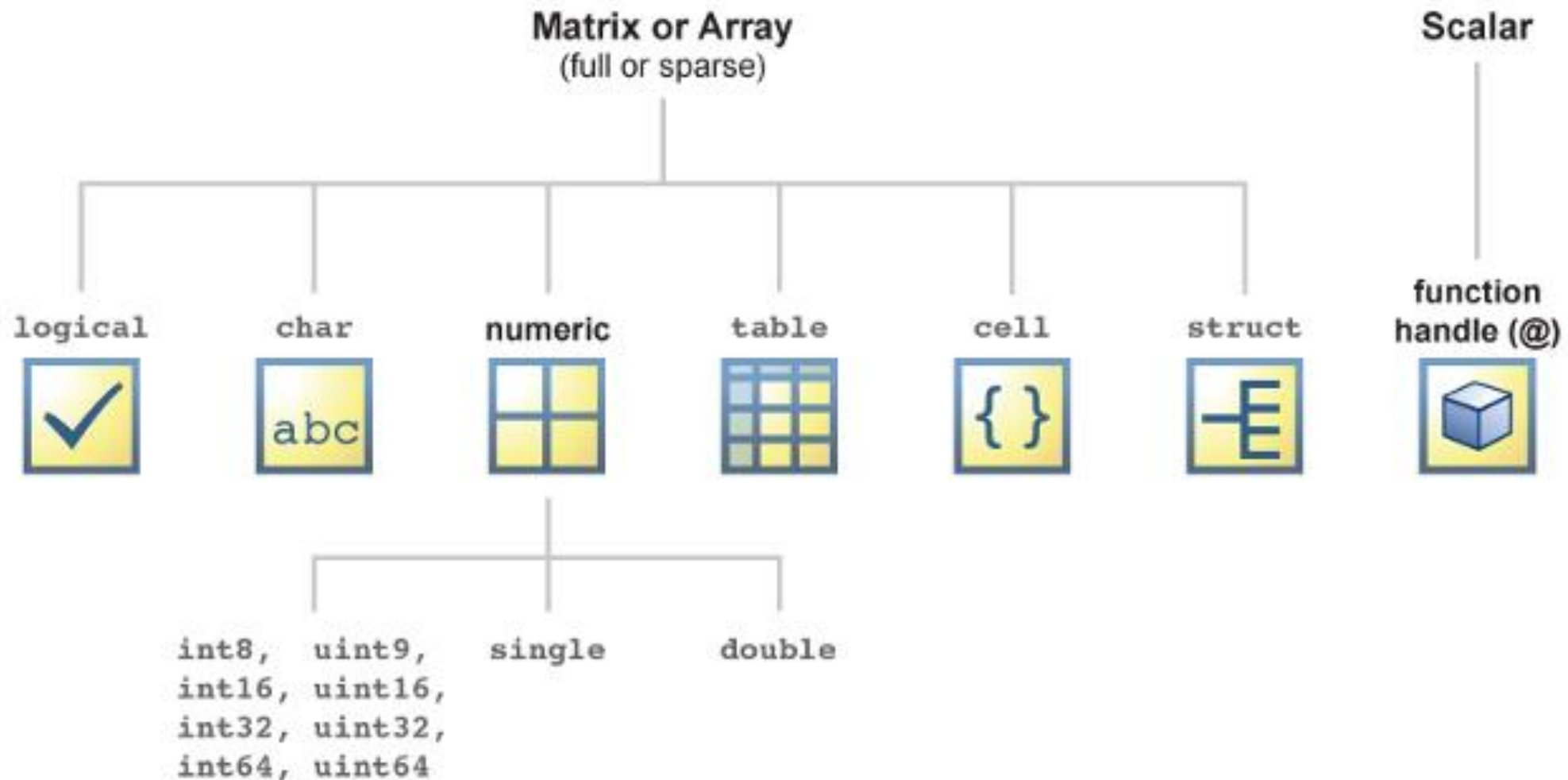
<https://www.ozgurzeydan.com/>



Değişken Adları

- Matlab'da değişken adları mutlaka harf ile başlar. Harf rakam veya alt çizgi (_) ile devam eder.
- Değişken adlarında sadece İngiliz alfabesindeki harfler kullanılır. Türkçe karakterler (ı, ç, ş, ü, ö, ğ) kullanılmaz.
- MATLAB büyük / küçük harf duyarlıdır. Örneğin:
 - **Data**, **data** ve **DATA** farklı değişkenlerdir.
- Geçerli değişken adları:
 - Ogresci
 - ogrenci_1
 - ogrenci2020
- Geçersiz değişken adları :
 - ogrenci-1
 - ogrenci.1
 - 1.ogrenci
 - Ogresci bir

Değişken Türleri





Değişken Sınırları

Tür	Maksimum Değer	Minimum Değer	Bayt
logical	1	0	1
int8	127	-128	1
int16	32767	-32768	2
int32	2.14e+09	-2.14e+09	4
int64	9.22e+18	-9.22e+18	8
uint8	255	0	1
uint16	65535	0	2
uint32	4.29e+09	0	4
uint64	1.84e+19	0	8
single	3.40e+038	-3.40e+038	4
double	1.79e+308	-1.79e+308	8

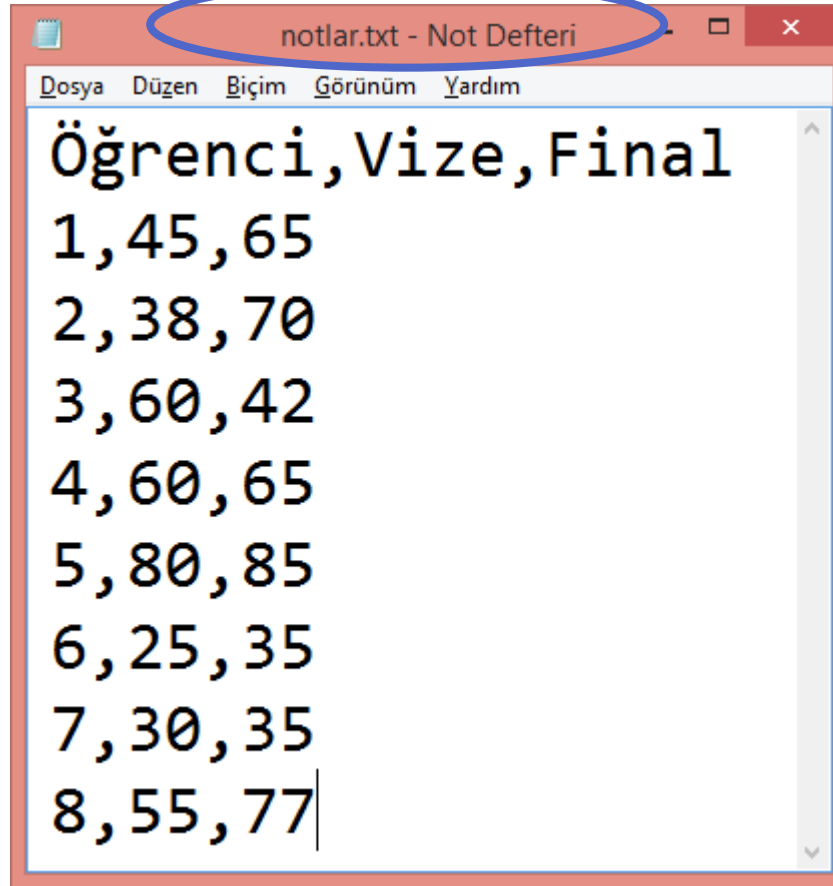
Veri Türleri

- `numara=input('Bir numara yazın: ')`
- **"numara" değişkeninin veri türü: double**
- `ad=input('Adınız nedir? ','s')`
- **"ad" değişkeninin veri türü: char**

```
>> numara=input('Bir numara yazın: ')
Bir numara yazın: 15|
numara =
    15
>> ad=input('Adınız nedir? ','s')
Adınız nedir? Özgür
ad =
Özgür
```

Workspace		
Name ▲	Value	Class
 ad	'Özgür'	char
 numara	15	double

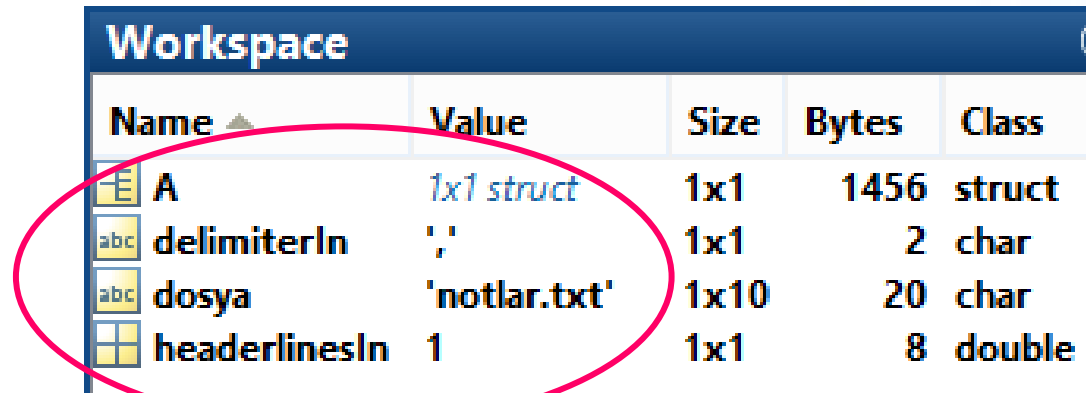
Text Dosyalarını Okuma



```
notlar.txt - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
Öğrenci,Vize,Final
1,45,65
2,38,70
3,60,42
4,60,65
5,80,85
6,25,35
7,30,35
8,55,77
```

Matlab'da Text Dosyalarının Okunması

- `dosya='notlar.txt';`
- `delimiterIn=',';`
- `headerlinesIn=1;`
- `A=importdata(dosya, delimiterIn, headerlinesIn);`



Workspace				
Name	Value	Size	Bytes	Class
A	1x1 struct	1x1	1456	struct
delimiterIn	','	1x1	2	char
dosya	'notlar.txt'	1x10	20	char
headerlinesIn	1	1x1	8	double

Dosyadan Okunan Veriler

```
>> A
A =
      data: [8x3 double]
  textdata: {'Öğrenci'  'Vize'  'Final'}
  colheaders: {'Öğrenci'  'Vize'  'Final'}
>> A.data
ans =
     1     45     65
     2     38     70
     3     60     42
     4     60     65
     5     80     85
     6     25     35
     7     30     35
     8     55     77
```

Değerlerin Değişkenlere Atanması

- 2. sütunu "**vize_notu**" ve 3. sütunu da "**final_notu**" değişkenine aktaralım:
- `vize_notu=A.data(:,2);`
- `final_notu=A.data(:,3);`

```
>> vize_notu=A.data(:,2)
vize_notu =
    45
    38
    60
    60
    80
    25
    30
    55
```

```
>> final_notu=A.data(:,3)
final_notu =
    65
    70
    42
    65
    85
    35
    35
    77
```

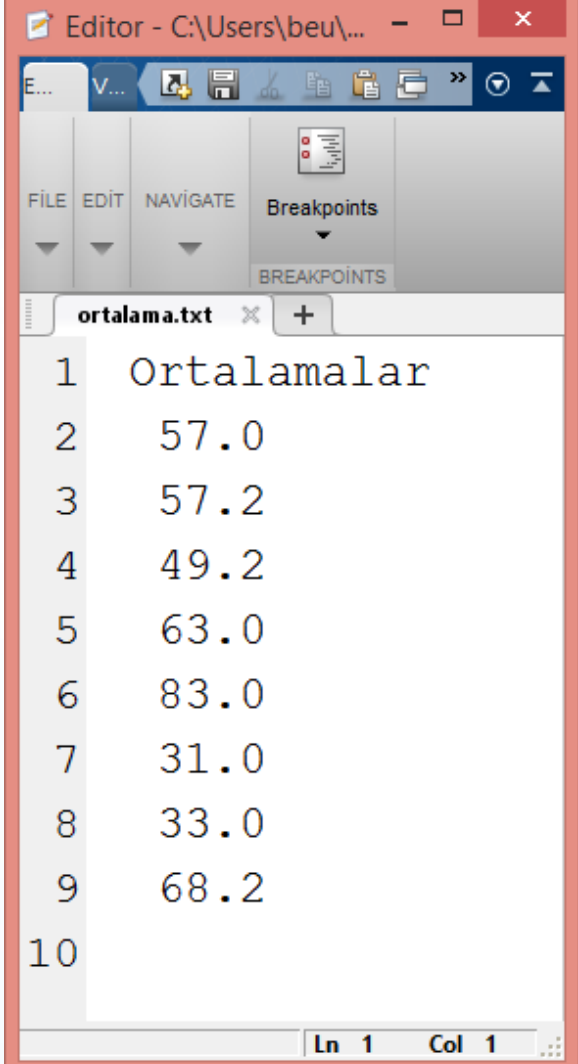
Okunan Veri İle İşlem Yapılması

- Vize notları %40 ve final notları da %60 olmak üzere ortalama not hesaplayalım.
- `ort=0.4*vize_notu+0.6*final_notu`

```
>> ort=0.4*vize_notu+0.6*final_notu
ort =
 57.0000
 57.2000
 49.2000
 63.0000
 83.0000
 31.0000
 33.0000
 68.2000
```

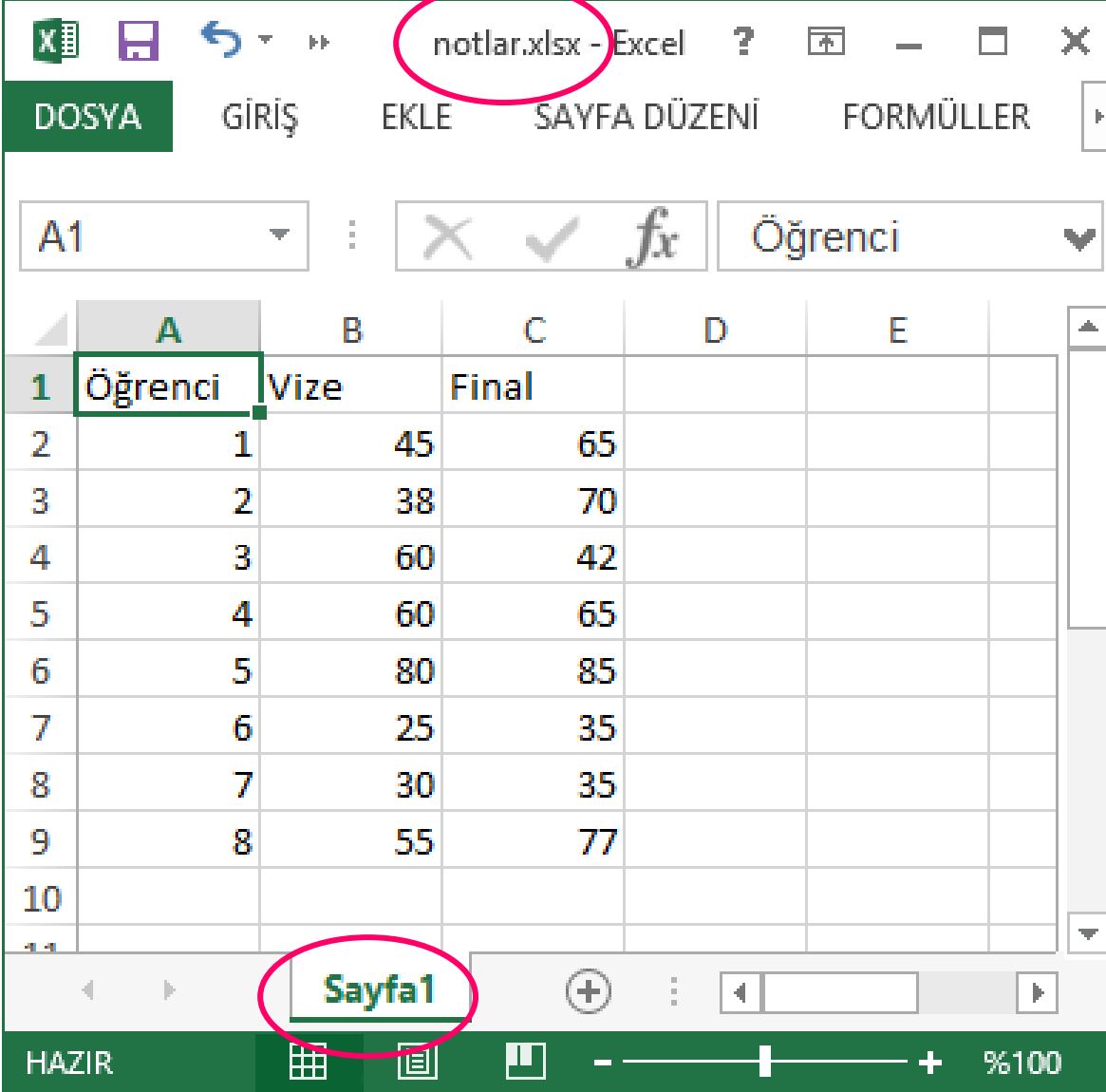
Text Dosyalarının Kaydedilmesi

- `fileID = fopen('ortalama.txt', 'w');`
- `fprintf(fileID, 'Ortalamalar\n');`
- `fprintf(fileID, '%5.1f\n', ort);`
- `fclose(fileID);`
- `\n` yeni satıra geçmek için kullanılır. Not defteri hariç diğer metin editörlerinde (WordPad, MS Word gibi) `\n` yeni satır olarak algılanır.
- `%5.1f` format belirlemek için kullanılır (Yazı 5 haneli yazılacak ondalıklı basamak 1 hane olacak).
- `%f`: double, `%d`:integer, `%s`: char



```
Editor - C:\Users\beu\...
FILE EDIT NAVIGATE Breakpoints
BREAKPOINTS
ortalama.txt x +
1 Ortalamalar
2 57.0
3 57.2
4 49.2
5 63.0
6 83.0
7 31.0
8 33.0
9 68.2
10
Ln 1 Col 1
```


MS Excel'den Veri Okunması



The screenshot displays the Microsoft Excel interface. The title bar at the top shows the file name 'notlar.xlsx - Excel'. The ribbon includes 'DOSYA', 'GİRİŞ', 'EKLE', 'SAYFA DÜZENİ', and 'FORMÜLLER'. The active cell is A1, containing the text 'Öğrenci'. The spreadsheet data is as follows:

	A	B	C	D	E
1	Öğrenci	Vize	Final		
2	1	45	65		
3	2	38	70		
4	3	60	42		
5	4	60	65		
6	5	80	85		
7	6	25	35		
8	7	30	35		
9	8	55	77		
10					
11					

The sheet name 'Sayfa1' is visible in the bottom status bar, and the zoom level is set to 100%.

MS Excel'den Veri Okunması

- **xlsread** fonksiyonu Excel'den veri okumak için kullanılır.
- `xlsread('file_name','sheet_name','range')`
- `vize=xlsread('notlar.xlsx','Sayfa1','B2:B9')`
- `final=xlsread('notlar.xlsx','Sayfa1','C2:C9')`

MS Excel'den Veri Okunması

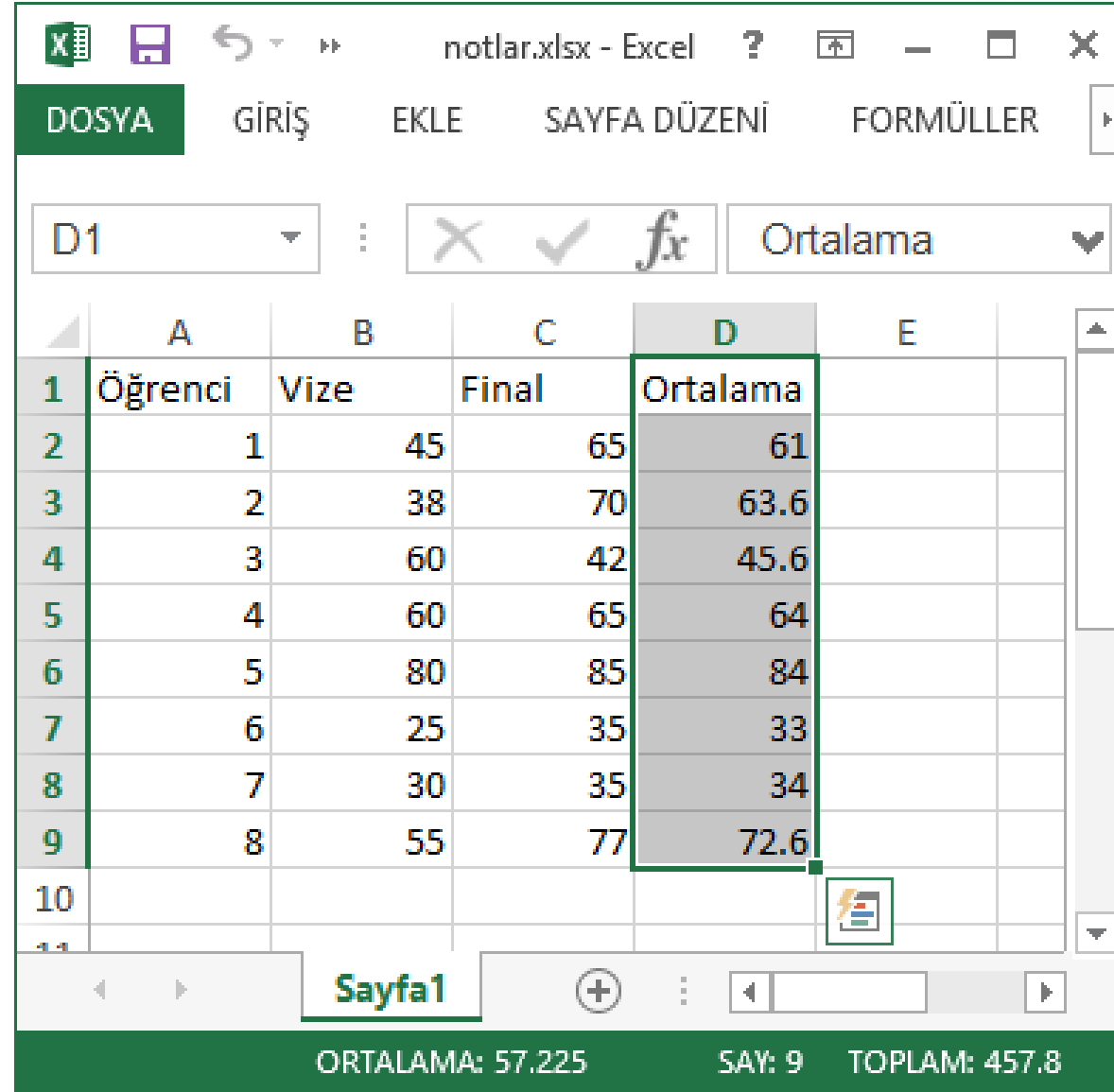
```
>> vize=xlsread('notlar.xlsx','Sayfa1','B2:B9')
vize =
    45
    38
    60
    60
    80
    25
    30
    55
```

```
>> final=xlsread('notlar.xlsx','Sayfa1','C2:C9')
final =
    65
    70
    42
    65
    85
    35
    35
    77
```

Excel'e Veri Kaydedilmesi

- Önce ortalamayı hesaplayalım.
- `ortalama=0.2*vize+0.8*final;`
- Ardından, **xlsxwrite** fonksiyonu ile hesapladığımız veriyi Excel'e kaydedelim.
- `xlsxwrite('file_name',variable_name,'Sheet','Range')`
- `xlsxwrite('notlar.xlsx',ortalama,'Sayfa1','D2:D9')`
- `header={'Ortalama'};`
- `xlsxwrite('notlar.xlsx',header,'Sayfa1','D1')`

Excel'e Veri Kaydedilmesi

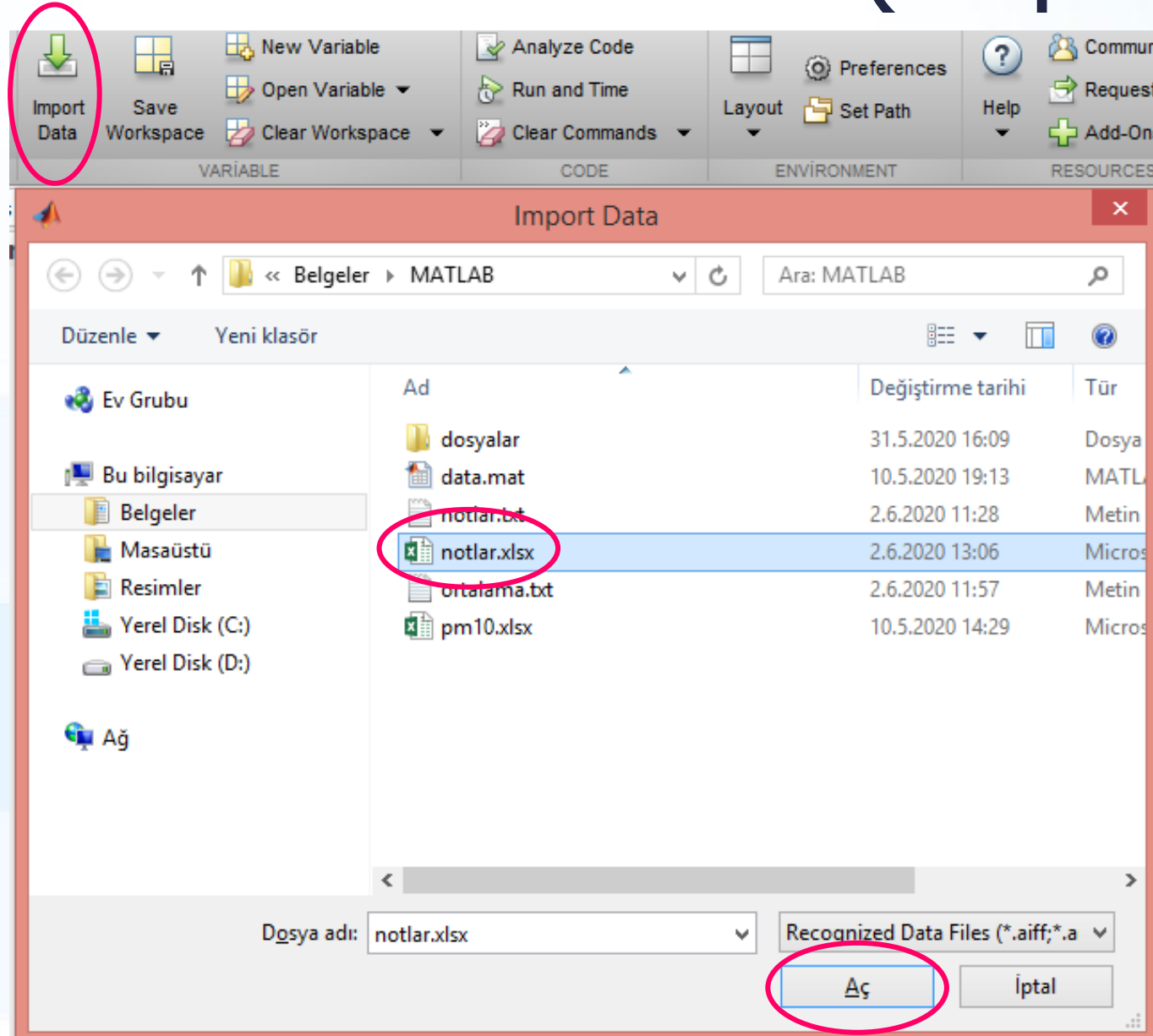


The screenshot shows the Microsoft Excel interface. The title bar indicates the file is named "notlar.xlsx". The ribbon is set to "FORMÜLLER" (Formulas). The formula bar shows the active cell is D1, containing the formula "Ortalama". The spreadsheet data is as follows:

	A	B	C	D	E
1	Öğrenci	Vize	Final	Ortalama	
2	1	45	65	61	
3	2	38	70	63.6	
4	3	60	42	45.6	
5	4	60	65	64	
6	5	80	85	84	
7	6	25	35	33	
8	7	30	35	34	
9	8	55	77	72.6	
10					
11					

The status bar at the bottom shows the following summary statistics: ORTALAMA: 57.225, SAY: 9, TOPLAM: 457.8.

Matlab'da Veri Aktarımı (Import Data)



Desteklenen Dosya Türleri

The screenshot displays the MATLAB R2013b environment. The 'Import Data' dialog box is open, showing the file 'notlar.xlsx' selected. The 'Recognized Data Files' list is visible, including various file formats such as Audio, CompuServe Graphics Interchange, Cursor Format, HDF, Icon Format, JPEG, MATLAB Data File, Portable Network Graphics, Spreadsheet, Tagged Image File Format, Text, Video, Windows or OS/2 Bitmap, and Zsoft Paintbrush. The Command Window shows the following MATLAB code:

```
>> xlswrite('notlar.xlsx', ortalama, 'Sayfal', 'D2:D9')
>> header={'Ortalama'};
>> xlswrite('notlar.xlsx', header, 'Sayfal', 'D1')
fx >>
```

The Workspace window shows the following variables:

Name	Value	Size	Bytes	Class
final	[65;70;42;6...	8x1	64	double
header	1x1 cell	1x1	128	cell
ortalama	[61;63.600...	8x1	64	double
vize	[45;38;60;6...	8x1	64	double

Matlab'da Veri Aktarımı

The screenshot displays the MATLAB Import Wizard window for the file 'notlar.txt'. The interface is divided into several sections: 'DELIMITERS', 'SELECTION', 'IMPORTED DATA', and 'IMPORT'. The 'DELIMITERS' section shows 'Delimited' selected with a 'Delimiter' dropdown. The 'SELECTION' section shows 'Range: A1:A9' and 'Variable Names Row: 1'. The 'IMPORTED DATA' section shows a list of data types: 'Column vectors', 'Matrix', 'Cell Array', and 'Table', with 'Matrix' selected. The 'IMPORT' section shows a green checkmark and 'Import Selection' button. The main workspace shows a table with the following data:

	A
	VarName1
	TEXT
1	Öğrenci,Vize,Fi...
2	1,45,65
3	2,38,70
4	3,60,42
5	4,60,65
6	5,80,85
7	6,25,35
8	7,30,35
9	8,55,77

Importing Data

The screenshot shows the MATLAB R2013b interface with the 'Import Data' dialog box open for the file 'notlar.txt'. The dialog is set to 'Delimited' format with 'Comma' as the column delimiter. The 'Range' is set to 'B2:C9' and 'Variable Names Row' is set to '1'. The 'Import Selection' button is highlighted. A table of data is visible in the dialog, and the Command History window shows the generated MATLAB code.

	A	B	C
	VarName1	Vize	Final
	NUMBER	NUMBER	NUMBER
1	Öğrenci	Vize	Final
2	1	45	65
3	2	38	70
4	3	60	42
5	4	60	65
6	5	80	85
7	6	25	35
8	7	30	35
9	8	55	77

```
vize=xlsread('notlar', 'Vize');
final=xlsread('notlar', 'Final');
ortalama=0.2*vize+0.8*final;
xlswrite('notlar.xlsx', 'Ortalama', ortalama);
clear
clc
```

Aktarılmak
istenen veri
seçilir.

Imported Data on Workspace

➤ **Sütun vektörü** formatında aktarım:

Workspace				
Name ▲	Value	Size	Bytes	Class
Final	[65;70;42;6...	8x1	64	double
Vize	[45;38;60;6...	8x1	64	double

➤ **Matris** formatında aktarım:

Workspace				
Name ▲	Value	Size	Bytes	Class
notlar	<i>8x2 double</i>	8x2	128	double



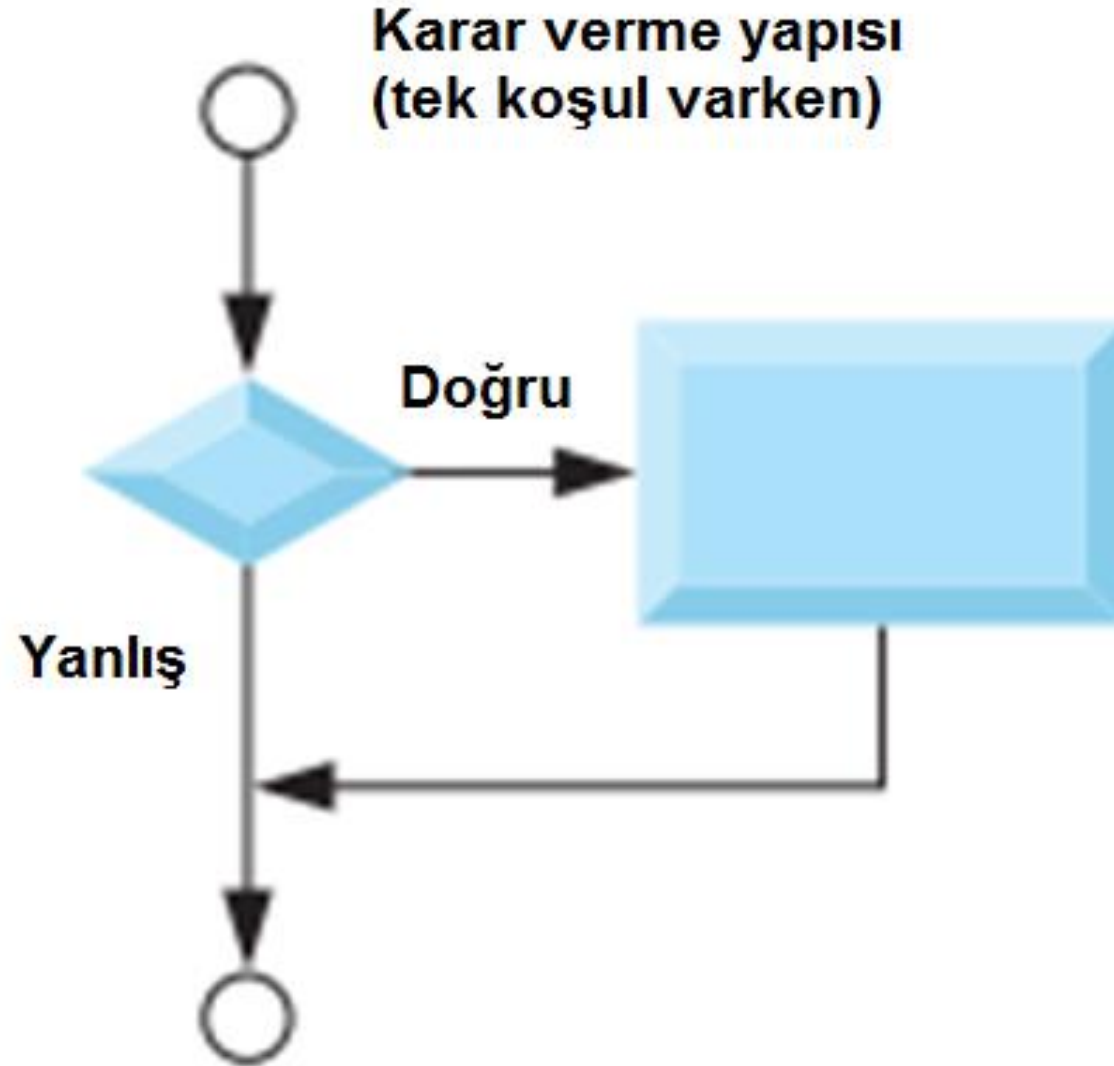
Matlab'da Karar Verme Yapıları

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Tek Koşullu Karar Verme Yapısı



if (Eğer) Komutu

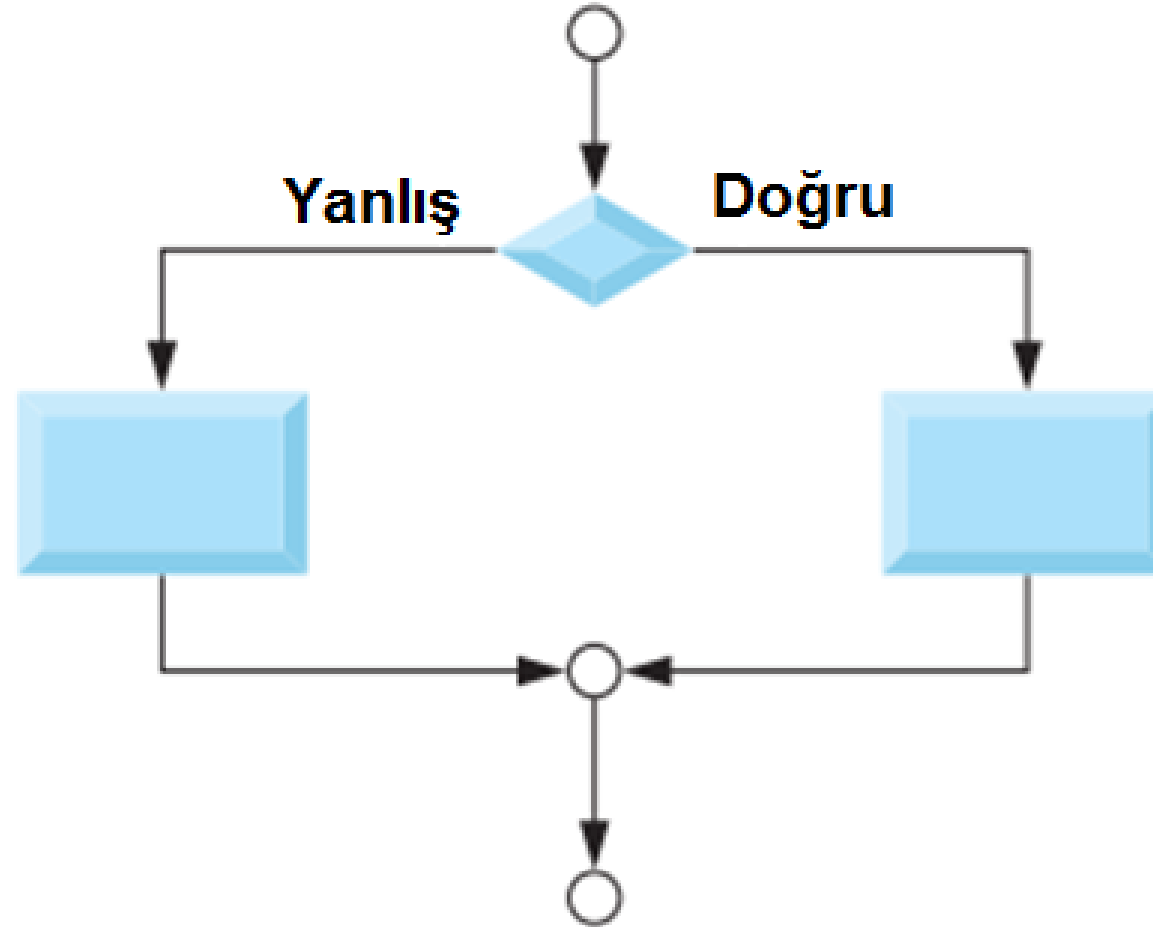
Herhangi bir koşul kontrol edilip doğru olması durumunda işlem yapılacaksa if (eğer) yapısı kullanılır.

```
if (koşul)
    Komutlar
end
```

```
>> x=9;
>> if (x<10) disp('x değeri 10 dan küçüktür.')
end
x değeri 10 dan küçüktür.
```

İki Koşullu Karar Verme Yapısı

Karar verme yapısı
(iki koşul varken)



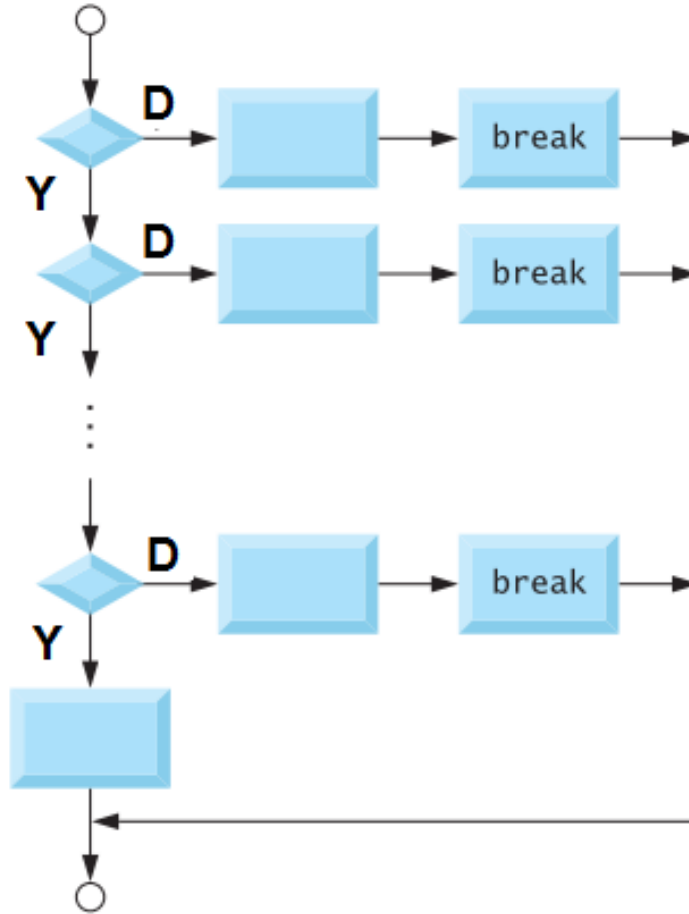
if (Eğer) Komutu

```
if (koşul)
    Komutlar-1
else
    Komutlar-2
end
```

```
>> x=9;
>> if (x<10) disp('x değeri 10 dan küçüktür.')
else disp('x değeri 10 dan büyüktür.')
end
x değeri 10 dan küçüktür.
```

Çok Koşullu Karar Verme Yapısı

Karar verme yapısı
(çok koşul varsa)



if (Eğer) Komutu

```
if (koşul-1)
    Komutlar-1
elseif (koşul-2)
    Komutlar-2
else
    Komutlar-3
end
```

```
>> x=9;
>> if (x<10) disp('x değeri 10 dan küçüktür.')
elseif (x>10) disp('x değeri 10 dan büyüktür')
else disp('x 10 a eşittir.')
end
x değeri 10 dan küçüktür.
```

İlişkisel Operatörler

Eşittir	$==$
Eşit değildir	\neq
Büyüktür	$>$
Küçüktür	$<$
Büyük veya Eşittir	\geq
Küçük veya Eşittir	\leq

Mantıksal Operatörler

Ve	&
Veya	
Değil	~

if Örneği - 1

- Verilen bir sayının tek veya çift olduğunu bulan programı yazınız.
- `sayi=23;`
- `if (rem(sayi,2) == 1) disp('Tek sayı')`
- `else disp('Çift sayı')`
- `end`
- **veya**
- `if (rem(sayi,2) ~= 1) disp('Çift sayı')`
- `else disp('Tek sayı')`
- `end`

if Örneđi - 2

- MATLAB'da if yapısını kullanarak notu verilen bir öğrencinin sene sonu karne harfini hesaplayan programı yazınız.

Harf	Not Aralığı
A	90 – 100
B	80 – 89
C	70 – 79
D	60 – 69
F	0 – 59

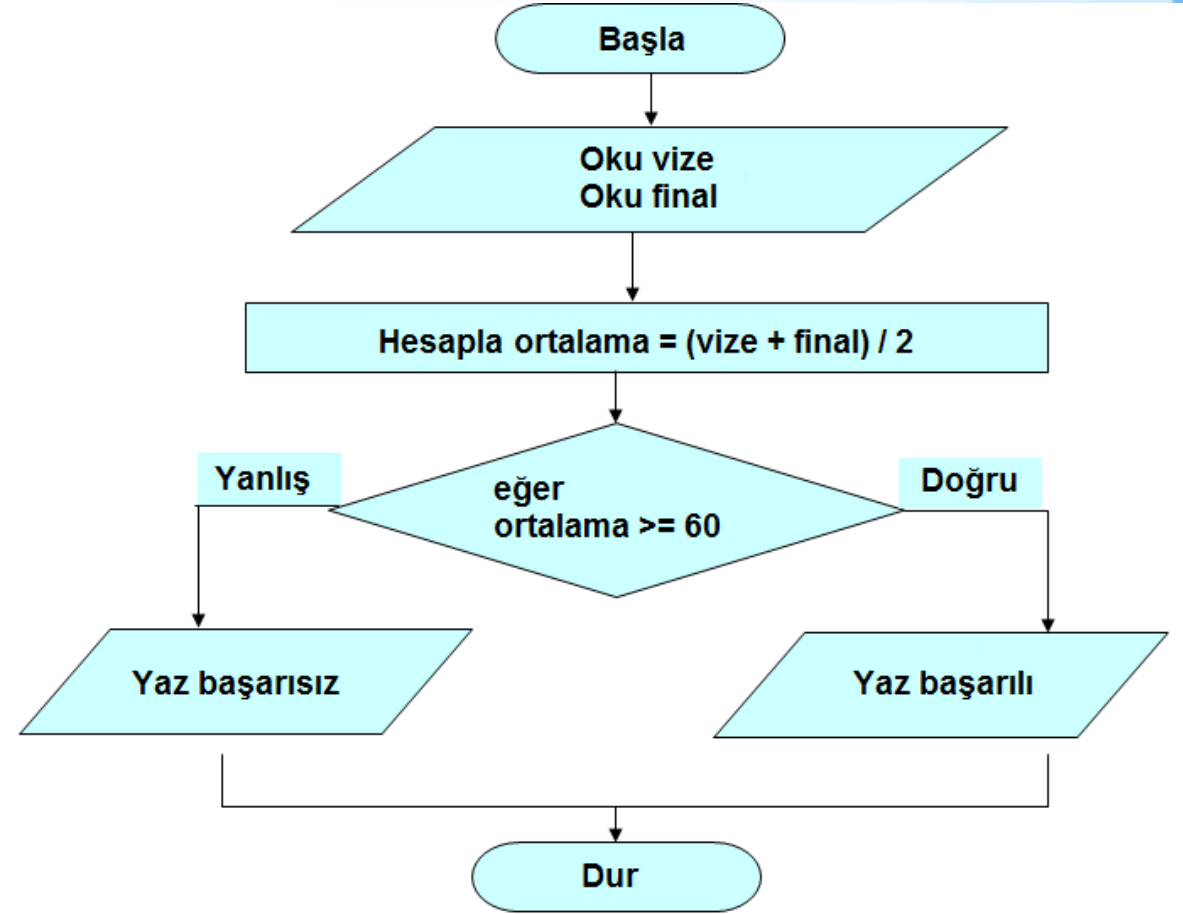
if Örneği - 2 - Cevap

- `if not >= 90 harf='A'`
- `elseif not >= 80 harf='B'`
- `elseif not >= 70 harf='C'`
- `elseif not >= 60 harf='D'`
- `else harf='F'`
- `end`

```
>> not=77;  
>> if not >= 90 harf='A'  
elseif not >= 80 harf='B'  
elseif not >= 70 harf='C'  
elseif not >= 60 harf='D'  
else harf='F'  
end  
harf =  
C
```

if Örneđi - 3

- Bir öğrencinin notu hesaplayan program yazınız. Program öğrencinin **vize** ve **final** notunu okuyarak **ortalamasını** hesaplayacaktır.
- Ortalamanın 60 veya daha fazla olması durumunda öğrenci başarılı sayılacaktır.
- $ortalama = (vize + final) / 2$



if Örneği - 3 - Cevap

- `vize=input('vize notu nedir? ');`
- `final=input('final notu nedir? ');`
- `ortalama=(vize+final)/2;`
- `if ortalama>=60`
`disp('Başarılı')`
- `else`
`disp('Başarısız')`
- `end`

```
>> vize=input('vize notu nedir? ');  
vize notu nedir? 45  
>> final=input('final notu nedir? ');  
final notu nedir? 70  
>> ortalama=(vize+final)/2;  
>> if ortalama>=60 disp('Başarılı')  
else disp('Başarısız')  
end  
Başarısız
```


Mantıksal Operatör İle if Kullanımı

- Örnek: ZBEÜ'de öğrencinin bağlı değerlendirmeye tabi olması için hem ortalamasının hem de final notunun en az 35 veya fazlası olması gerekmektedir.

Eğer (**final** ≥ 35 **ve** **ortalama** ≥ 35)

Yaz "**bağlı sisteme tabi**"

Değilse

Yaz "**Notu: FF**"

Eğeri bitir.

- `if (final \geq 35 & ortalama \geq 35)`
- `disp('Bağlı sisteme tabi')`
- `else`
- `disp('Not: FF')`
- `end`

İç İçe if (Nested if) Kullanımı

- Örnek: ZBEÜ'de öğrencinin bağıl değerlendirmeye tabi olması için hem ortalamasının hem de final notunun en az 35 veya fazlası olması gerekmektedir.

Eğer **final** ≥ 35 ise

{ eğer **ortalama** ≥ 35 ise

Yaz "**bağıl sisteme tabi**"

Değilse

Yaz "**Notu: FF**"

eğeri bitir }

Değilse

Yaz "**Notu: FF**"

Eğeri bitir.

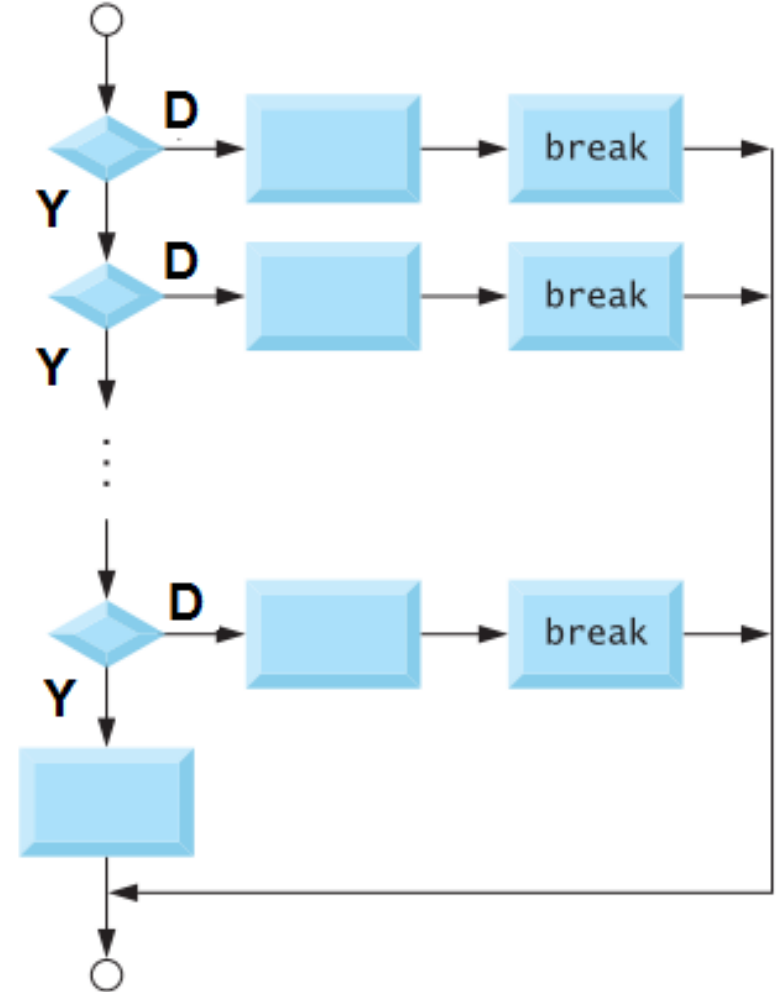
İç İçe if (Nested if) Kullanımı

```
if (final>=35)
    if (ortalama>=35)
        disp('Bağlı sisteme tabi')
    else
        disp('Notu: FF')
    end
else
    disp('Notu: FF')
end
```

Switch Komutu

- `switch` durum
- `case` koşul-1, komutlar-1
- `case` koşul-2, komutlar-2
- `case` koşul-3, komutlar-3
- ...
- `case` koşul-n, komutlar-n
- `otherwise` komutlar-n+1
- `end`

Karar verme yapısı
(çok koşul varsa)



Switch Örneği - 1

- `gun=input('Bugün haftanın hangi günü? ');`
- `Bugün haftanın hangi günü? 2`
- `switch gun`
- `case 1, disp('Pazartesi')`
- `case 2, disp('Salı')`
- `case 3, disp('Çarşamba')`
- `case 4, disp('Perşembe')`
- `case 5, disp('Cuma')`
- `case 6, disp('Cumartesi')`
- `case 7, disp('Pazar')`
- `otherwise disp('Yanlış yazdınız!')`
- `end`
- `Salı`

Switch Örneği - 2

- Matlab'da **rand** and **floor** fonksiyonlarını kullanarak zar oyunu programı yazalım.
- Her 2 zar da 1 ile 6 arasında rastgele sayı üretsın:
- `zar1 = floor(6*rand) + 1;`
- `zar2 = floor(6*rand) + 1;`
- İki zarın toplamı 7 ise oyun kazanılsın, aksi halde kaybedilsin.
- Bu koşulun kontrolünü **switch** ile yapalım.



Switch Örneği - 2 - Cevap

- `zar1 = floor(6*rand) + 1;`
- `zar2 = floor(6*rand) + 1;`
- `switch zar1+zar2`
- `case 7, disp([num2str(zar1+zar2), ' -
Tebrikler kazandınız.'])`
- `otherwise disp([num2str(zar1+zar2), ' -
Kaybettiniz.'])`
- `end`

Switch Örneği - 2 - Cevap

```
>> zar1 = floor(6*rand) + 1;
zar2 = floor(6*rand) + 1;
switch zar1+zar2
case 7, disp([num2str(zar1+zar2), ' - Tebrikler kazandınız.'])
otherwise disp([num2str(zar1+zar2), ' - Kaybettiniz.'])
end
7 - Tebrikler kazandınız.
```

```
>> zar1 = floor(6*rand) + 1;
zar2 = floor(6*rand) + 1;
switch zar1+zar2
case 7, disp([num2str(zar1+zar2), ' - Tebrikler kazandınız.'])
otherwise disp([num2str(zar1+zar2), ' - Kaybettiniz.'])
end
5 - Kaybettiniz.
```



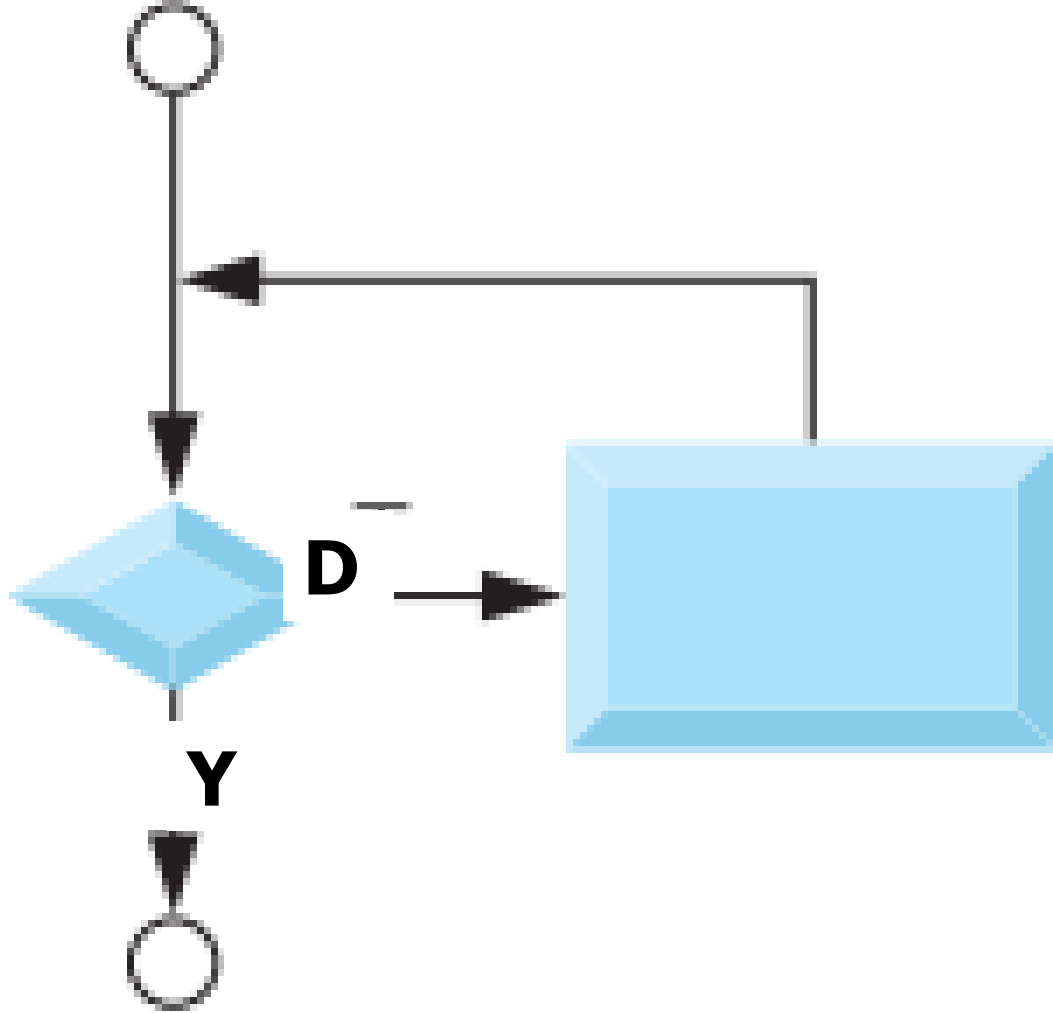

Matlab'da Döngü Yapıları

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Döngüler (Tekrarlar)



- Programlama sırasında aynı işlem birden çok defa tekrarlanacaksa döngüler vasıtasıyla yapılırlar.
- Matlab'da kullanılan döngü komutları **for** ve **while** döngüleridir.

For Döngüsü

- For döngüsünde bilinen bir sayıda tekrarlanacak işlemler yaptırılır.

- Döngü yapısı:

```
for döngü değişkeni
```

```
    Komutlar
```

```
end
```

Yandaki örnekte **i** değişkeni sayaç olarak adlandırılır.

```
>> for i=1:5
disp('for döngüsü örneği')
end
for döngüsü örneği
for döngüsü örneği
for döngüsü örneği
for döngüsü örneği
for döngüsü örneği
```

For Örneği -1

- 1'den 10'a kadar olan sayıların karelerinin toplamını hesaplan bir programı Matlab'da for döngüsü ile yazalım.

$$\sum_{i=1}^{10} i^2$$

```
>> toplam=0;  
>> for i=1:10  
toplam=toplam+i*i;  
end  
>> toplam  
toplam =  
    385
```

For Örneđi - 2

- 5 kişilik bir sınıftaki öğrencilerin not ortalamasını **for** döngüsü kullanarak hesaplayan bir **script** yazınız.
- Girdiler: 5 öğrencinin notu
- İşlemler:
 - Notların toplamını hesapla
 - Öğrenci sayısını say
 - Ortalamayı hesapla, **ortalama = toplam not / sayaç**
- Çıktı: not ortalaması

For Örneği - 2 - Cevap

```
toplam_not=0;
for sayac=1:5
    not=input([num2str(sayac), '. öğrencinin notu:
    ']);
    toplam_not=toplam_not+not;
end
ortalama=toplam_not/sayac;
disp(['Sınıf ortalaması: ',num2str(ortalama)])
```

Scripti ortalama1.m adı ile kaydedelim.

For Örneği - 2 - Ekran Çıktısı

```
>> ortalama1  
1. öğrencinin notu: 15  
2. öğrencinin notu: 60  
3. öğrencinin notu: 43  
4. öğrencinin notu: 87  
5. öğrencinin notu: 64  
Sınıf ortalaması: 53.8
```

Bilgisayar Hafızasındaki Değişkenler

	not	toplam_not	sayaç	ortalama
Döngü öncesi		0	0	
Döngü sırasında	45	45	1	
	60	105	2	
	75	180	3	
	100	280	4	
	25	305	5	
Döngü sonrasında		305	5	61

Notların sırasıyla 45, 60, 75, 100 ve 25 olarak girildiği düşünüldüğünde.

İç İçe Döngüler

- Bazı durumlarda döngülerin iç içe kullanılması gerekir.
- Bu durumda içteki döngünün sayacı tümünden tamamlanmadan dıştaki döngünün sayaç değeri artmaz.

➤ Örnek kullanım:

```
for i=1:n % dış döngü
    for j=1:m % iç döngü
        komutlar
    end % iç döngü sonu
end % dış döngü sonu
```

İç İçe Döngüler - Örnek

- Bir matrisin her bir elemanının değerini kullanıcıdan isteyen ve sonrasında önce matrisin tamamını sonra da her bir elemanını teker teker ekrana yazdıran bir script yazınız.
- Kolaylık olması için 2x3 boyutunda bir matris oluşturalım.
- Toplam döngü sayısı: $2 \times 3 = 6$

İç İçe Döngüler - Matlab Kodu

- `disp('Bu script 2x3 matris değerlerini kullanıcından ister.')`
- `for i=1:2`
- `for j=1:3`
- `veri(i,j)=input([num2str(i),'. satır',num2str(j),'. sütun değeri yaz: ']);`
- `end`
- `end`
- `disp('Oluşturulan matris:')`
- `veri`

İç İçe Döngüler - Matlab Kodu (devamı)

```
➤ disp('Matrisin her bir değeri:')
➤ for i=1:2
➤     for j=1:3
➤         disp([num2str(i),'. satır
',num2str(j),'. sütun değeri:
',mat2str(veri(i,j))]);
➤         %disp()
➤     end
➤ end
```

İç İçe Döngüler - Çıktı Ekranı

Bu script 2x3 matris değerlerini kullanıcıdan ister.

1. satır 1. sütun değeri yaz: 0

1. satır 2. sütun değeri yaz: 1

1. satır 3. sütun değeri yaz: 2

2. satır 1. sütun değeri yaz: 3

2. satır 2. sütun değeri yaz: 4

2. satır 3. sütun değeri yaz: 5

Oluşturulan matris:

veri =

0 1 2

3 4 5

Matrisin her bir değeri:

1. satır 1. sütun değeri: 0

1. satır 2. sütun değeri: 1

1. satır 3. sütun değeri: 2

2. satır 1. sütun değeri: 3

2. satır 2. sütun değeri: 4

2. satır 3. sütun değeri: 5

While Döngüsü

- **While** döngüsü kontrol edilen koşul doğru olduğu sürece sürekli olarak çalışacaktır.

```
while koşul
    Komutlar
end
```

- Programcı sonsuz döngü oluşmaması için koşulun ne zaman sonlanacağını kontrol etmelidir.
- Sonsuz döngü oluşursa **Ctrl + Break** kombinasyonu ile çalıştırdığınız kodları sonlandırabilirsiniz.

```
>> j=5;
while j>0
disp(j)
j=j-1;
end

    5

    4

    3

    2

    1
```

While Örnekleri

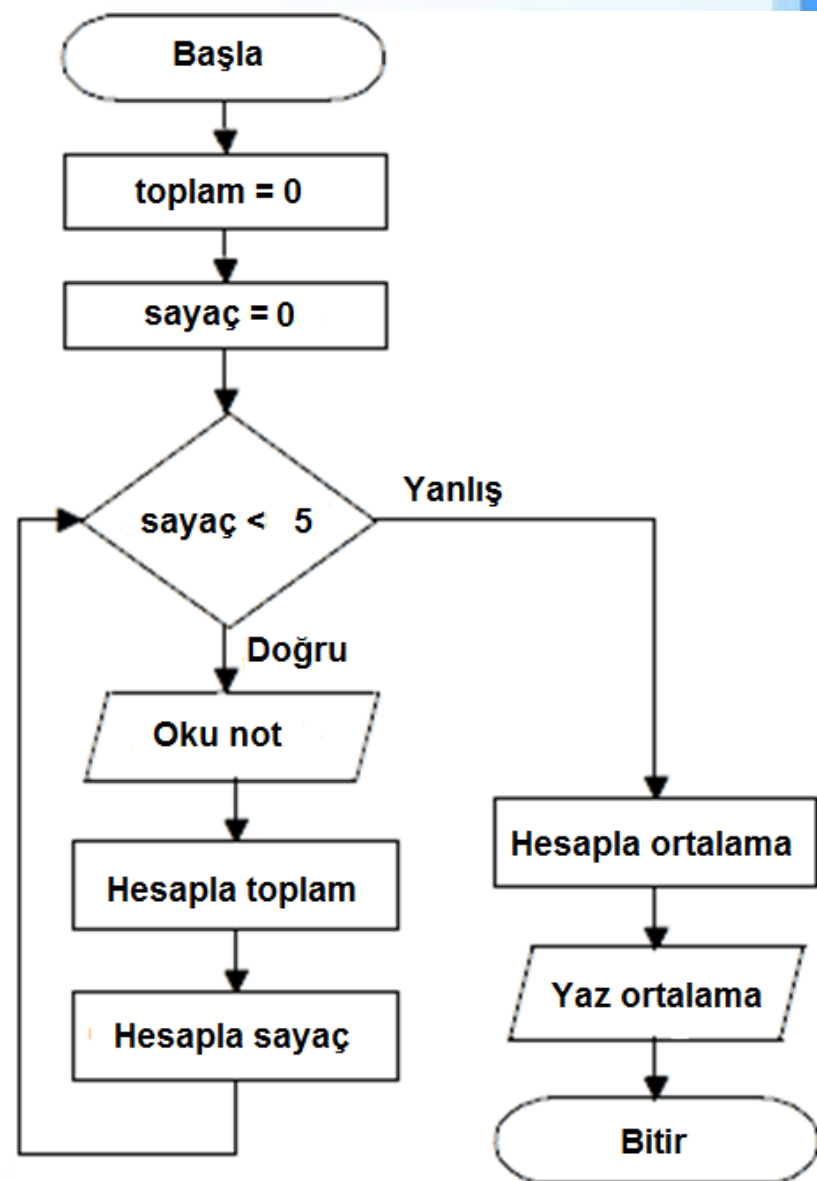
- "For Örneği - 2"de verilen problemi **while** döngüsü kullanarak tekrar çözüyoruz.
- İlk uygulamada sayaç = 0 olsun. Döngü sayaç < 5 olduğu sürece çalışsın.
- İkinci uygulamada sayaç = 1 olsun. Döngü sayaç <= 5 olduğu sürece çalışsın.

While Örneği - 1

- `toplam_not=0;`
- `sayac=0;`
- `while sayac<5`
- `not=input([num2str(sayac+1), '. öğrencinin`
`notu: ']);`
- `toplam_not=toplam_not+not;`
- `sayac=sayac+1;`
- `end`
- `ortalama=toplam_not/sayac;`
- `disp(['Sınıf ortalaması: ', num2str(ortalama)])`

While Örneđi - 1

```
>> ortalama1  
1. öğrencinin notu: 10  
2. öğrencinin notu: 20  
3. öğrencinin notu: 30  
4. öğrencinin notu: 40  
5. öğrencinin notu: 50  
Sınıf ortalaması: 30
```

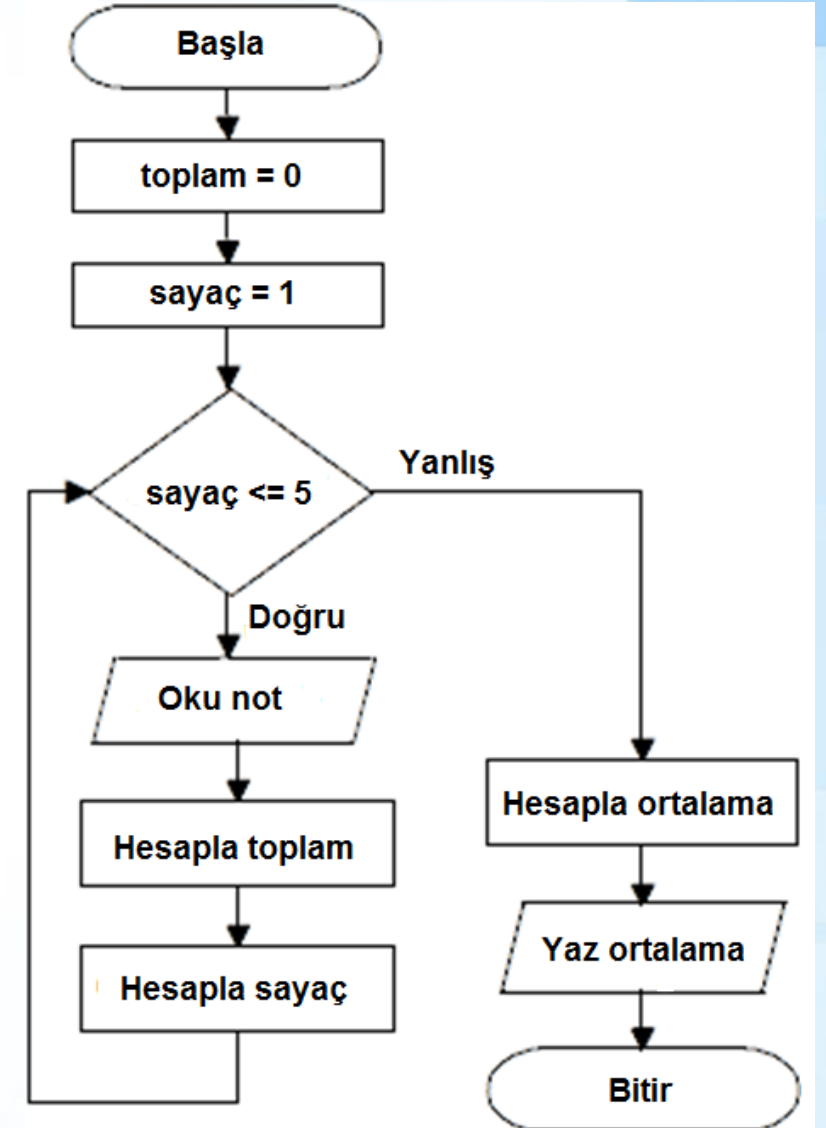


While Örneği - 2

- `toplam_not=0;`
- `sayac=1;`
- `while sayac<=5`
- `not=input([num2str(sayac), '. öğrencinin`
`notu: ']);`
- `toplam_not=toplam_not+not;`
- `sayac=sayac+1;`
- `end`
- `ortalama=toplam_not/(sayac-1);`
- `disp(['Sınıf ortalaması: ', num2str(ortalama)])`

While Örneđi - 2

```
>> ortalama3  
1. öğrencinin notu: 20  
2. öğrencinin notu: 40  
3. öğrencinin notu: 60  
4. öğrencinin notu: 80  
5. öğrencinin notu: 100  
Sınıf ortalaması: 60
```



While Örneđi - 3

- Öğrenci sayısı bilinmeyen bir sınıftaki öğrencilerin not ortalamasını hesaplayan bir program yazınız.
- Öğrenci sayısı bilinmediđi için **sentinel kontrolü** kullanılmalıdır.
- örneđin **not = -99** yazılırsa döngü sonlansın.

While Örneği - 3 - Cevap

```
➤ toplam_not=0;
➤ sayac=0;
➤ not=0;
➤ while (not~-99)
➤     not=input([num2str(sayac+1), '. öğrencinin notu: ']);
➤     if (not~-99)
➤         toplam_not=toplam_not+not;
➤         sayac=sayac+1;
➤     end
➤ end
➤ ortalama=toplam_not/(sayac);
➤ disp(['Sınıf ortalaması: ', num2str(ortalama)])
```

While Örneği - 3 - Ekran Çıktısı

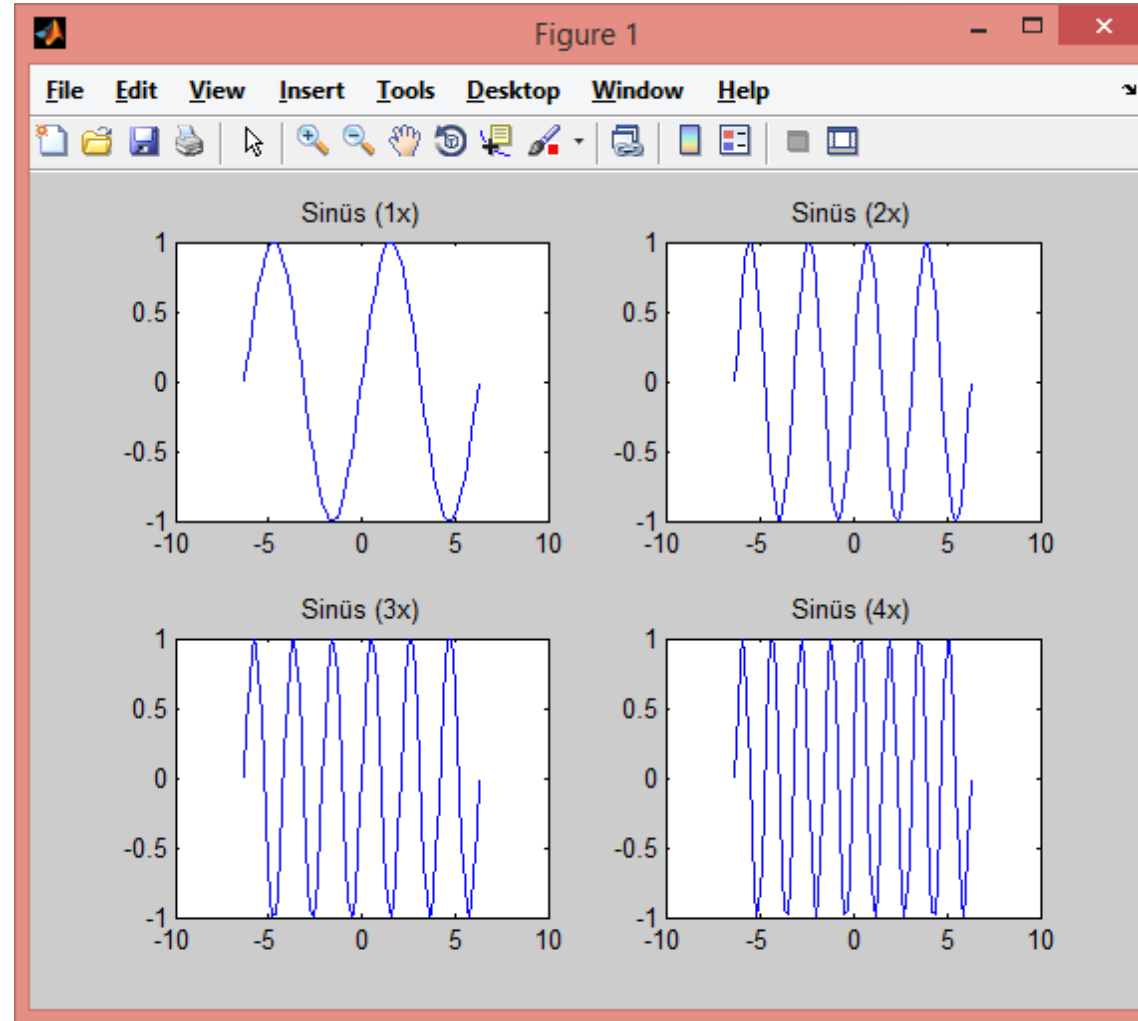
```
1. öğrencinin notu: 20  
2. öğrencinin notu: 40  
3. öğrencinin notu: 60  
4. öğrencinin notu: 80  
5. öğrencinin notu: -99  
Sınıf ortalaması: 50
```

While Örneği - 4

- **while** döngüsü ve **subplot** komutlarını kullanarak $\sin(x)$, $\sin(2x)$, $\sin(3x)$, $\sin(4x)$ grafiklerini bir arada çizdirelim.

```
x=linspace(-2*pi,2*pi,100);  
i=1;  
while i<=4  
    subplot(2,2,i)  
    plot(x,sin(i*x))  
    title(['Sinüs (' ,num2str(i) , 'x) '])  
    i=i+1;  
end
```

While Örneği - 4 - Grafik Çıktısı





Matlab'da Döngü Yapıları (2)

Hata Ayıklama, Break, Continue

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>

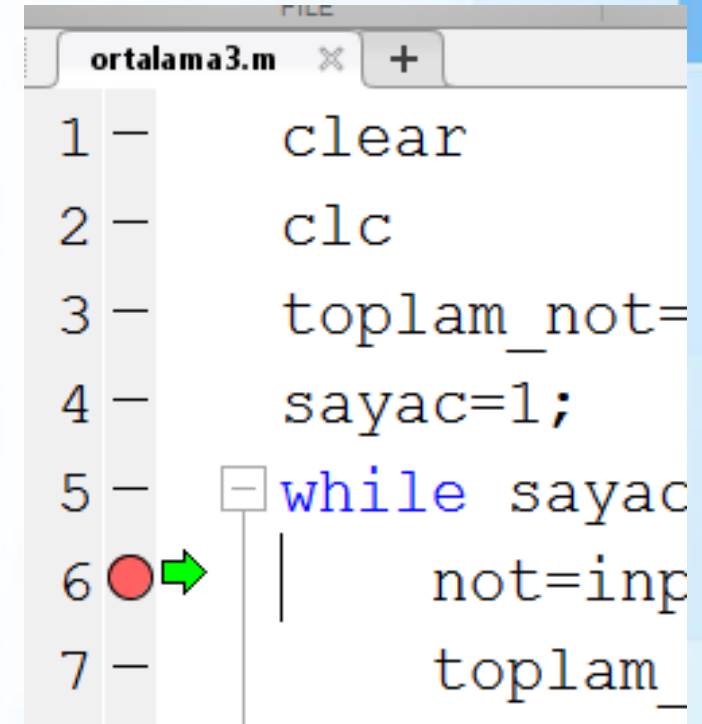


Hata Ayıklama (Debugging)

- Programlama sırasında 3 tür hata ile karşılaşabilirsiniz:
 1. **Yazım Hatası** (syntax error): Bu durumda Matlab kodu çalıştırmayacak ve hata uyarısı verecektir.
 2. **Runtime hatası**: Programın çalışması sırasında hata oluşurmasıdır.
 3. **Mantık Hatası** (logic error): Programda yazım ve runtime hatası olmaması, programın düzgün çalışması fakat yanlış sonuçlar vermesidir. Programcı kodlama yaparken mantıksal bir hata yapmıştır.
- Hata ayıklama yapabilmek için kodların satır satır çalıştırılması ve Workspace'de yer alan değişkenlerin değerlerinin nasıl değiştiğinin gözlenmesi faydalı olabilir.

Hata Ayıklama (Debugging)

- Hata ayıklamayı açabilmek için editör üzerinde satır numarasının sağındaki çizginin üzerine tıklanıp kırmızı nokta (**Breakpoint**) eklenir.
- Script bu şekilde çalıştırıldığında bu noktada çalışması duracak ve kullanıcıdan gelecek diğer komutu bekleyecektir.
- **Continue** düğmesine basıldığında bir sonraki Breakpoint'e kadar kodlar çalıştırılır.
- **Step** düğmesine basıldığında kodlar satır satır çalıştırılır.



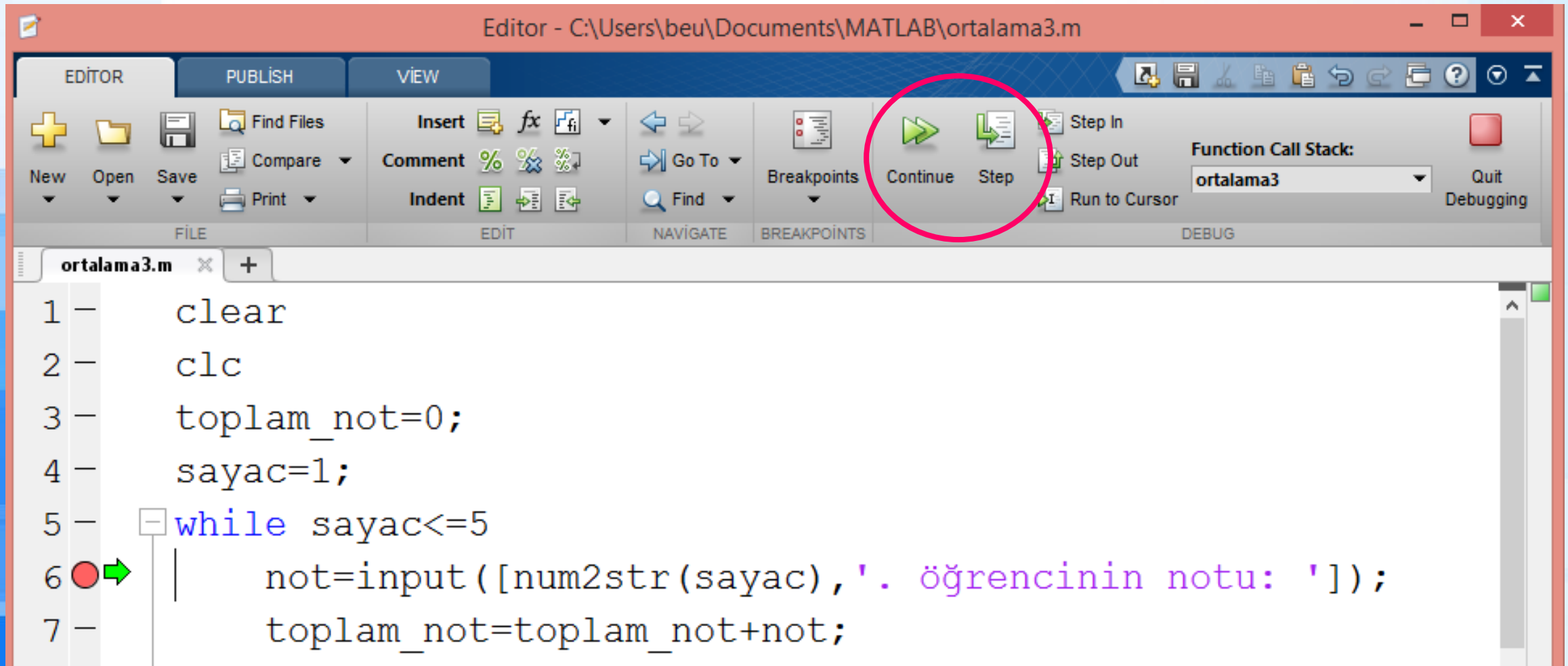
```
FILE
ortalama3.m x +
1 - clear
2 - clc
3 - toplam_not=
4 - sayac=1;
5 - while sayac
6 - | not=inp
7 - | toplam_
```

The screenshot shows a MATLAB script editor window titled 'ortalama3.m'. The script contains the following code:

```
1 - clear
2 - clc
3 - toplam_not=
4 - sayac=1;
5 - while sayac
6 - | not=inp
7 - | toplam_
```

A red circle (breakpoint) is placed on the left margin of line 6, with a green arrow pointing to it. A vertical line extends from the breakpoint to the start of the code on line 6.

Continue ve Step Komutları



The screenshot shows the MATLAB Editor interface with the following components:

- Editor Tabs:** EDITOR, PUBLISH, VIEW.
- File Menu:** New, Open, Save, Find Files, Compare, Print.
- Edit Menu:** Insert, Comment, Indent, Go To, Find.
- Breakpoints Menu:** Breakpoints.
- Debug Menu:** Continue (highlighted with a red circle), Step, Step In, Step Out, Run to Cursor, Quit Debugging.
- Function Call Stack:** ortalama3.

The code editor displays the following MATLAB code:

```
1 - clear
2 - clc
3 - toplam_not=0;
4 - sayac=1;
5 - while sayac<=5
6 -     not=input([num2str(sayac), '. öğrencinin notu: ']);
7 -     toplam_not=toplam_not+not;
```

A red circle highlights the 'Continue' and 'Step' buttons in the Debug menu. A red circle and a green arrow point to the 'while' loop in the code editor.

Workspace'deki Değerlerin Gözlenmesi

- Script Debugging modunda çalıştırılırken Workspace'deki değerlerin değişimi gözlenerek programdaki mantık hataları tespit edilebilir.

Name ▲	Value	Size	Bytes	Class
not	65	1x1	8	double
sayac	2	1x1	8	double
toplam_not	55	1x1	8	double

Name ▲	Value	Size	Bytes	Class
not	65	1x1	8	double
sayac	2	1x1	8	double
toplam_not	120	1x1	8	double

Name ▲	Value	Size	Bytes	Class
not	65	1x1	8	double
sayac	3	1x1	8	double
toplam_not	120	1x1	8	double

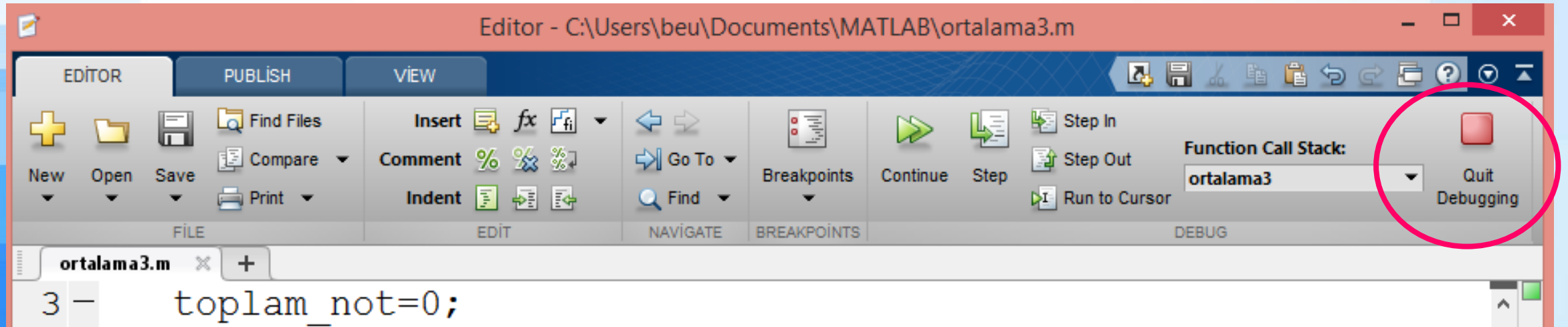
Name ▲	Value	Size	Bytes	Class
not	67	1x1	8	double
sayac	3	1x1	8	double
toplam_not	187	1x1	8	double

Name ▲	Value	Size	Bytes	Class
not	67	1x1	8	double
sayac	4	1x1	8	double
toplam_not	187	1x1	8	double

Name ▲	Value	Size	Bytes	Class
not	23	1x1	8	double
sayac	5	1x1	8	double
toplam_not	210	1x1	8	double

Hata Ayıklamadan Çıkılması

- Editör üzerindeki Quit Debugging düğmesine basılarak hata ayıklamadan çıkılır.



Break Komutu

- **Break** komutu döngüyü sonlandırmak için kullanılır.
- Yandaki 2 örnek arasındaki farka dikkat ediniz.

```
>> for i=1:5  
disp(i)  
if i==3 break  
end  
end  
1  
2  
3
```

```
>> for i=1:5  
if i==3  
break  
end  
disp(i)  
end  
1  
2
```

Continue Komutu

- **Continue** komutu, döngü içerisinde mevcut kodun çalıştırılmadan bir sonraki sayaç değeri için döngünün tekrar çalıştırılmasını sağlar.
- Yandaki 2 örnek arasındaki farka dikkat ediniz.

```
>> for i=1:5  
if i==3  
continue  
end  
disp(i)  
end
```

1
2
4
5

```
>> for i=1:5  
disp(i)  
if i==3  
continue  
end  
end
```

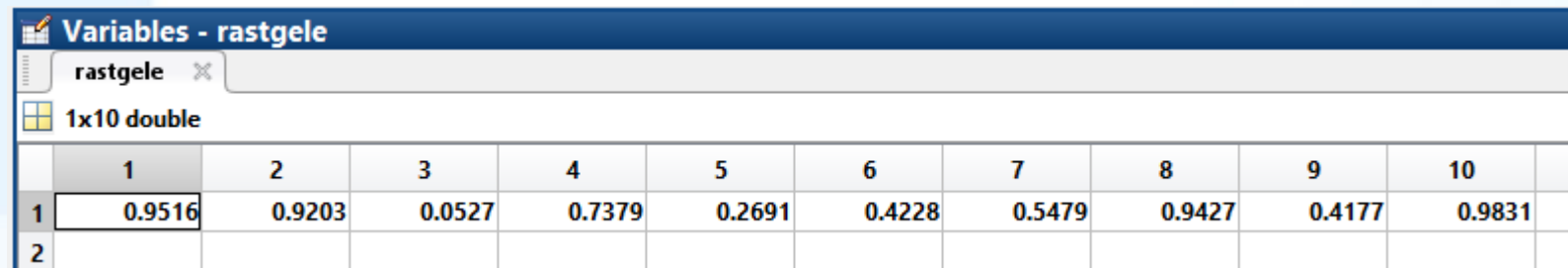
1
2
3
4
5

Döngü Kullanımından Kaçınmak

- **find** fonksiyonu ile döngü kullanmak yerine daha basit kod yazarak işlem yapmak mümkündür
- find fonksiyonu istenilen kritere uyan değerlerin bulunduğu matristeki **indis** değerlerini verecektir.

find Komutu

- Örnek: rand fonksiyonu ile 10 elemanlı rastgele sayılardan oluşan bir matris oluşturalım.
- `rastgele=rand(1,10)`
- Değeri 0.5'den daha büyük olan sayıların indislerini find komutu ile bulabiliriz:
- `find(rastgele>0.5)`



The screenshot shows the MATLAB Variables window for a variable named 'rastgele'. It is a 1x10 double array. The values are: 0.9516, 0.9203, 0.0527, 0.7379, 0.2691, 0.4228, 0.5479, 0.9427, 0.4177, 0.9831.

	1	2	3	4	5	6	7	8	9	10
1	0.9516	0.9203	0.0527	0.7379	0.2691	0.4228	0.5479	0.9427	0.4177	0.9831
2										

```
>> find(rastgele>0.5)
ans =
     1     2     4     7     8    10
```

Döngü Yerine find Komutu Kullanmak

- **rastgele** matrisinin içinde yer alan 0.5'den büyük değerlerin sayısını tespit etmek isteyelim.
- Bu işlem için bir döngü komutu (**for** veya **while**) ve bir de koşul komutu (**if**) gereklidir.

```
say=0;  
for i=1:10  
    if rastgele(i)>0.5  
        say=say+1;  
    end  
end
```

Döngü Yerine find Komutu Kullanmak

- Aynı işlemi find komutu yapalım. Find ile indis değerlerini bulduğumuz değişkenin boyutunu hesaplatmak yeterli olacaktır:
- `say2=length(find(rastgele>0.5))`

```
>> rastgele=rand(1,10);  
>> say=0;  
>> for i=1:10  
if rastgele(i)>0.5  
say=say+1;  
end  
end  
>> say  
say =  
     6  
  
>> say2=length(find(rastgele>0.5))  
say2 =  
     6
```



Programlama Uygulamaları

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Örnek - 1 - Geometrik Ortalama

- Kullanıcı tarafından girilecek sayıların geometrik ortalamasını hesaplayan Matlab scripti yazınız.
 1. Script kullanıcıya kaç tane sayı (**n**) olduğunu soracak
 2. Döngü **n** kere çalıştırılarak kullanıcıdan sayıları (**sayi**) isteyecek
 3. Girilen sayıların her biri çarpım (**carpim**) değişkeni ile çarpılacak.
 4. Sayı giriş işlemi bittikten sonra `nthroot` fonksiyonu ile sonuç (**sonuc**) hesaplanacak, ekrana yazdırılacak.
- Script **geo_ort.m** adı ile kaydedilsin.

Örnek - 1 - Cevap

- `n=input('Kaç tane sayı var? ');`
- `carpim=1;`
- `for sayac=1:n`
- `sayi(sayac)=input('Sayıyı yazın: ');`
- `carpim=carpim*sayi(sayac);`
- `end`
- `sonuc=nthroot(carpim,n);`
- `disp(['Geometrik ortalama: ',num2str(sonuc)])`

Örnek - 1 - Ekran Çıktısı

```
>> geo_ort  
Kaç tane sayı var? 3  
Sayıyı yazın: 2  
Sayıyı yazın: 4  
Sayıyı yazın: 8  
Geometrik ortalama: 4
```


Baloncuk Sıralama

Baloncuk Sıralama (Bubble Sort)

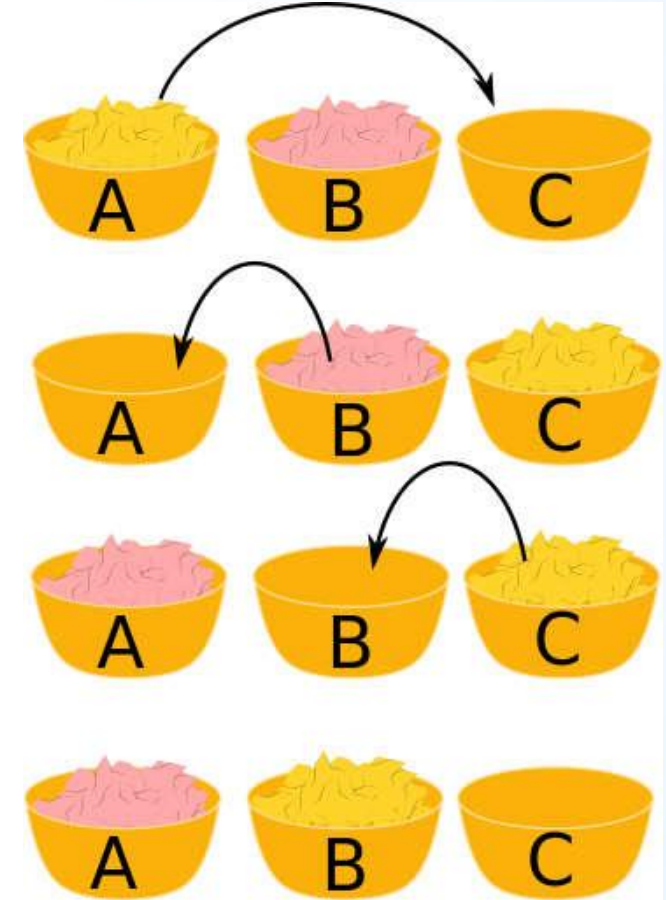
- En basit sıralama algoritmasıdır.
- Yan yana iki değerin birbiri ile kıyaslanıp gerekli durumda yerlerinin değiştirilmesi prensibine dayanır.

Animasyon

6 5 3 1 8 7 2 4

Değişken Değerlerinin Yer Değiştirmesi

- İki değerin yer değiştirmesi için üçüncü bir geçici değişken (temporary variable) kullanılır:
- $C=A$
- $A=B$
- $B=C$



Örnek - 2 - Baloncuk Sıralama

- Kullanıcı tarafından girilecek sayıları küçükten büyüğe Baloncuk sıralama algoritması kullanarak sıralayan Matlab scripti yazınız.
 1. Script kullanıcıya kaç tane sayı (**n**) olduğunu soracak.
 2. Döngü **n** kere çalıştırılarak kullanıcıdan sayıları (**sayi**) isteyecek.
 3. Girilen sayılar önce karışık olarak ekrana yazdırılacak.
 4. Baloncuk sıralama algoritması ile iki değer birbiri ile kıyaslanacak. İlk değer ikinciden büyükse değerler yer değiştirecek.
 5. Son olarak sıralanmış değerler ekrana yazdırılacak.
- Bu scriptin yazımında **sort()** fonksiyonu kullanılmayacak!

Örnek - 2 - Cevap

- `clear`
- `clc`
- `n=input('Kaç tane sayı var? ');`
- `for i=1:n %sayıların kullanıcıdan istenmesi`
- `sayi(i)=input('Sayıyı yazın: ');`
- `end`
- `disp(['Karışık olarak sayılar: ',
num2str(sayi)])`

Örnek - 2 - Cevap (devamı)

```
➤ for j=1:n %Baloncuk sıralama
➤   for k=1:n-1
➤     if sayi(k) > sayi(k+1)
➤       temp=sayi(k+1); %temp: geçici değişken
➤       sayi(k+1)=sayi(k);
➤       sayi(k)=temp;
➤     end
➤   end
➤ end
➤ disp(['Sıralı sayılar: ', num2str(sayi)])
```

Örnek - 2 - Ekran Çıktısı

```
Kaç tane sayı var? 6
```

```
Sayıyı yazın: 45
```

```
Sayıyı yazın: 23
```

```
Sayıyı yazın: -9
```

```
Sayıyı yazın: 0
```

```
Sayıyı yazın: 68
```

```
Sayıyı yazın: 27
```

```
Karışık olarak sayılar: 45 23 -9 0 68 27
```

```
Sıralı sayılar: -9 0 23 27 45 68
```

Örnek - 3 - Standart Sapma Programı

- Kullanıcı tarafından girilecek sayıların standart sapmasını hesaplayan Matlab scripti yazınız.

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

- Bu scriptin yazımında **std()**, **sum()** ve **mean()** fonksiyonları kullanmayınız.

Örnek - 3 - Program Tasarımı

- Hesaplamayı 2 boyutlu bir matris tasarlayarak yapalım:
- veri(n,3)
- Kullanıcı tarafından girilen sayılar ilk sütunda tutulsun: veri(n,1)
- Her bir sayı ile ortalamamanın farkı ikinci sütunda tutulsun: veri(n,2)
- İkinci sütundaki değerin karesi üçüncü sütunda tutulsun: veri(n,3)

i	x_i	$x_i - x_{\text{ort}}$	$(x_i - x_{\text{ort}})^2$
1			
2			
3			
4			
...			
...			
n-2			
n-1			
n			

Örnek - 3 - Cevap

- `clear`
- `clc`
- `n=input('Kaç tane sayı var? ');`
- `toplam1=0;`
- `for i=1:n`
- `sayi(i,1)=input('Sayıyı yazın: ');`
- `toplam1=toplam1+sayi(i,1);`
- `end`
- `ortalama=toplam1/n;`

Örnek - 3 - Cevap (devamı)

- `toplam2=0;`
- `for i=1:n`
- `sayi(i,2)=sayi(i,1)-ortalama;`
- `sayi(i,3)=sayi(i,2)*sayi(i,2);`
- `toplam2=toplam2+sayi(i,3);`
- `end`
- `S=sqrt(toplam2/(n-1));`
- `disp(['Standart Sapma: ', num2str(S)])`

Örnek - 3 - Ekran Çıktısı

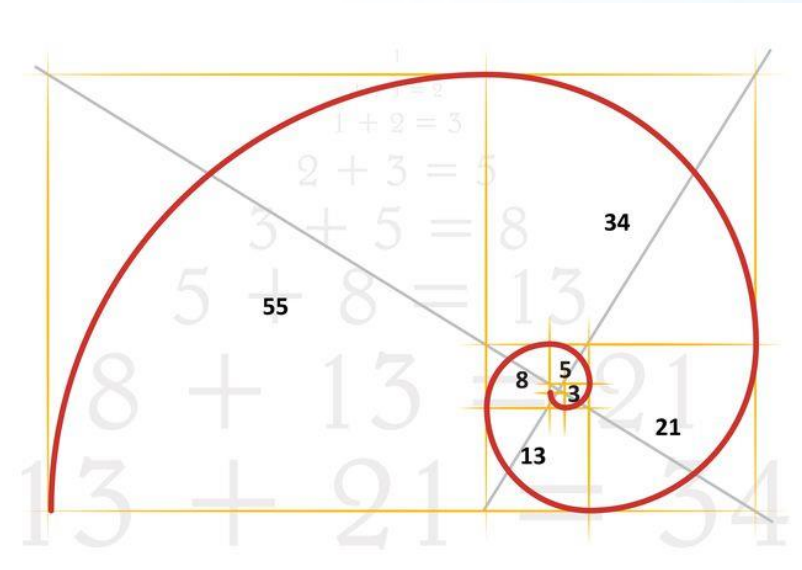
```
Kaç tane sayı var? 5  
Sayıyı yazın: 10  
Sayıyı yazın: 11  
Sayıyı yazın: 13  
Sayıyı yazın: 16  
Sayıyı yazın: 20  
Standart Sapma: 4.062
```

Kontrol:

```
>> std([10,11,13,16,20])  
ans =  
    4.0620
```

Örnek - 4 - Fibonacci Serisi

- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ... şeklinde yazılan ilk iki terim hariç diğer terimleri $f(n)=f(n-1)+f(n+2)$ formülü ile hesaplanan seriye Fibonacci Serisi denir. $f(0)=0$ ve $f(1)=1$ 'dir.
- Kullanıcı tarafından eleman sayısı girilen bir Fibonacci Serisi'nin elemanlarını yazdıran bir Matlab scripti yazınız.



Örnek - 4 - Cevap

```
➤ clear
➤ clc
➤ n=input('Kaç terim olacak? ');
➤ fib(1)=0;
➤ fib(2)=1;
➤ for i=3:n
➤     fib(i)=fib(i-1)+fib(i-2);
➤ end
➤ disp([num2str(n), ' terimli Fibonacci Serisi: ']);
➤ disp(num2str(fib(1:n)));
```

Örnek - 4 - Ekran Çıktısı

```
Kaç terim olacak? 13
```

```
13 terimli Fibonacci Serisi:
```

```
0    1    1    2    3    5    8    13    21    34    55    89    144
```

Örnek - 5 - Fibonacci Serisi'nin Terimleri

- Bir önceki örnek üzerinde değişiklikler yaparak Fibonacci Serisi'nin n.ci terimini while döngüsü ile hesaplayan fonksiyon yazalım.
- Fonksiyonun örnek girdisi: `fobonacci_n(10)`
- Fonksiyonun çıktısı: n.ci Fibonacci Terimi ekrana yazdırılsın.

Örnek - 5 - Cevap

```
➤ function fibonacci_n(n)
➤ %fibonacci_n fonksiyonu
  Fibonacci
➤ %Serisinin n.ci terimini
  hesaplar.
➤ f(1)=0;
➤ f(2)=1;
➤ if n==0
    disp(num2str(f(1)));
➤ elseif n==1;
    disp(num2str(f(2)));
➤ else
    i=3;
    while i<=n
        f(i)=f(i-1)+f(i-
    2);
        i=i+1;
    end
    disp(num2str(f(n)));
➤ end
➤ end
```


Örnek - 5 - Ekran Çıktısı

```
>> help fibonacci_n
fibonacci_n fonksiyonu Fibonacci
Serisinin n.ci terimini hesaplar.

>> fibonacci_n(2)
1
>> fibonacci_n(5)
3
>> fibonacci_n(10)
34
```



Programlama Uygulamaları (2)

© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Örnek - 6 - En Büyük Sayı

- Kullanıcı tarafından girilecek sayıların içerisinde en büyük olanı bulup ekrana yazdıran bir script yazınız.
 1. Script kullanıcıya kaç tane sayı (**n**) olduğunu soracak.
 2. Döngü **n** kere çalıştırılarak kullanıcıdan sayıları (**sayi**) isteyecek.
 3. Script `en_buyuk=-Inf;` olarak başlayacak.
 4. Kullanıcı tarafından girilen sayı **en_buyuk** değişkeninin değerinden daha büyük ise bu sayı **en_buyuk** değişkenine atanacak.
- Bu scriptin yazımında **mak()** fonksiyonunu kullanmayınız.

Örnek - 6 - Cevap

```
➤ clear sayi
➤ n=input('Kaç tane sayı var? ');
➤ en_buyuk=-Inf;
➤ for i=1:n
➤     sayi(i)=input('Sayıyı yazın: ');
➤     if sayi(i)>en_buyuk
➤         en_buyuk=sayi(i);
➤     end
➤ end
➤ disp(en_buyuk)
```

Örnek - 6 - Ekran Çıktısı

```
>> enbuyuk
Kaç tane sayı var? 5
Sayıyı yazın: 92
Sayıyı yazın: -123
Sayıyı yazın: 0
Sayıyı yazın: 45
Sayıyı yazın: 236
En büyük sayı: 236
```

Örnek - 7 - En Küçük Sayı

- Bir önceki örnekten faydalanarak, kullanıcı tarafından girilecek sayıların içerisinde en büyük olanı bulup ekrana yazdıran bir script yazınız.
- Bu scriptin yazımında **min()** fonksiyonunu kullanmayınız.

Örnek - 7 - Cevap

```
➤ clear sayi
➤ n=input('Kaç tane sayı var? ');
➤ en_kucuk=Inf;
➤ for i=1:n
➤     sayi(i)=input('Sayıyı yazın: ');
➤     if sayi(i)<en_kucuk
➤         en_kucuk=sayi(i);
➤     end
➤ end
➤ disp(['En küçük sayı: ',num2str(en_kucuk)])
```

Örnek - 7 - Ekran Çıktısı

```
>> enkucuk
```

```
Kaç tane sayı var? 4
```

```
Sayıyı yazın: -12
```

```
Sayıyı yazın: 15
```

```
Sayıyı yazın: 45
```

```
Sayıyı yazın: -17
```

```
En küçük sayı: -17
```


Örnek - 8 - Beden Kitle İndeksi Hesaplama

- Kullanıcı tarafından girilecek kilo (kg) ve boy (m) değerlerini kullanarak beden kitle indeksini (bki) hesaplayan bir Matlab fonksiyonu yazınız.
- $bki = \text{kilo} / (\text{boy}^2)$
- Fonksiyon yandaki tabloya göre kişinin durumunu yazacak, ayrıca normal kilo aralığını da hesaplayıp yazacaktır.

BKİ	Durum
<18.5	Zayıf
$\geq 18.5 - < 25$	Normal
$\geq 25 - < 30$	Hafif Şişman
$\geq 30 - < 35$	Şişman
≥ 35	Obez

Örnek - 8 - Cevap

- `function` `beden_kitle(kilo,boy)`
- `%beden_kitle(kilo,boy)` fonksiyonu `kilo` ve `boy` değerlerine göre
- `%Beden Kitle İndeksini hesaplar. Kilo (kg) ve boy (m) olarak yazılır.`
- `%Fonksiyon ayrıca normal kilo aralığını da verir.`
- `bki=kilo/(boy*boy);`
- `normal_alt=18.5*boy*boy;`
- `normal_ust=25*boy*boy;`

Örnek - 8 - Cevap (devamı)

```
➤ if bki<18.5
➤     disp('Zayıf')
➤ elseif bki<25
➤     disp('Normal kilo')
➤ elseif bki<30
➤     disp('Hafif kilolu')
➤ elseif bki<35
➤     disp('Şişman')
```

Örnek - 8 - Cevap (devamı)

```
➤ else
➤     disp('Obez')
➤ end
➤ fprintf('Normal kilo aralığınız: %4.1f -
%4.1f \n',normal_alt,normal_ust)
➤ end
```

Örnek - 8 - Ekran Çıktısı

```
>> help beden_kitle
```

```
beden_kitle(kilo,boy) fonksiyonu kilo ve boy değerlerine göre  
Beden Kitle İndeksini hesaplar. Kilo (kg) ve boy (m) olarak yazılır.  
Fonksiyon ayrıca normal kilo aralığını da verir.
```

```
>> beden_kitle(80,1.74)
```

```
Hafif kilolu
```

```
Normal kilo aralığınız: 56.0 - 75.7
```



Matlab'da Diyalog Pencereleeri

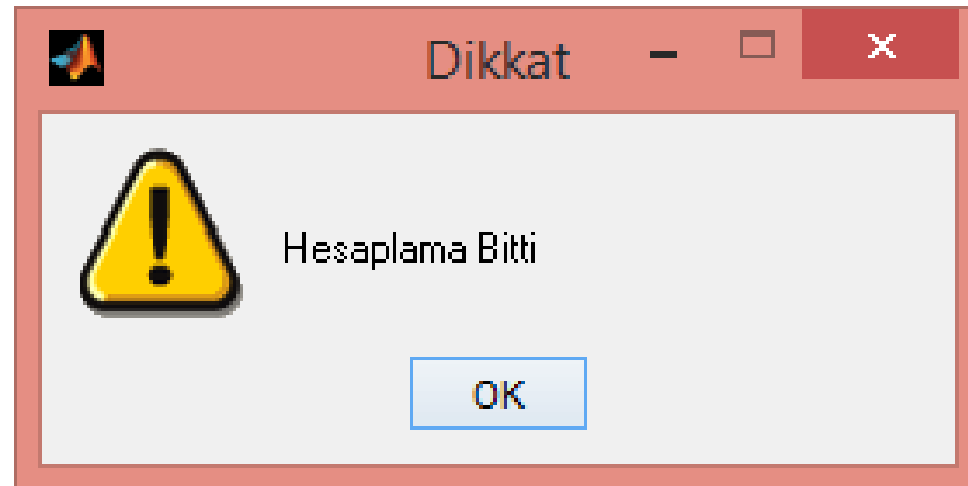
© Doç. Dr. Özgür ZEYDAN

<https://www.ozgurzeydan.com/>



Mesaj Kutusu

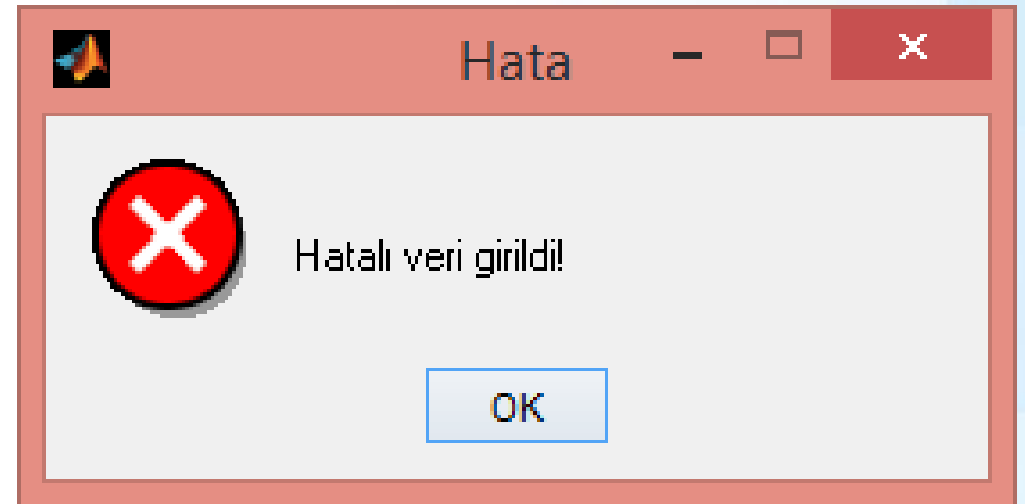
- İkaz, uyarı veya hata durumlarında ekranda beliren mesaj kutusu yardımıyla kullanıcıyı uyarmak için kullanılır.
- `msgbox(message, title, icon)`
- `msgbox('Hesaplama Bitti', 'Dikkat', 'warn')`



Mesaj Kutusu - İkon Türleri

Özellik	İkon
'help'	
'warn'	
'error'	
'none'	İkon yok

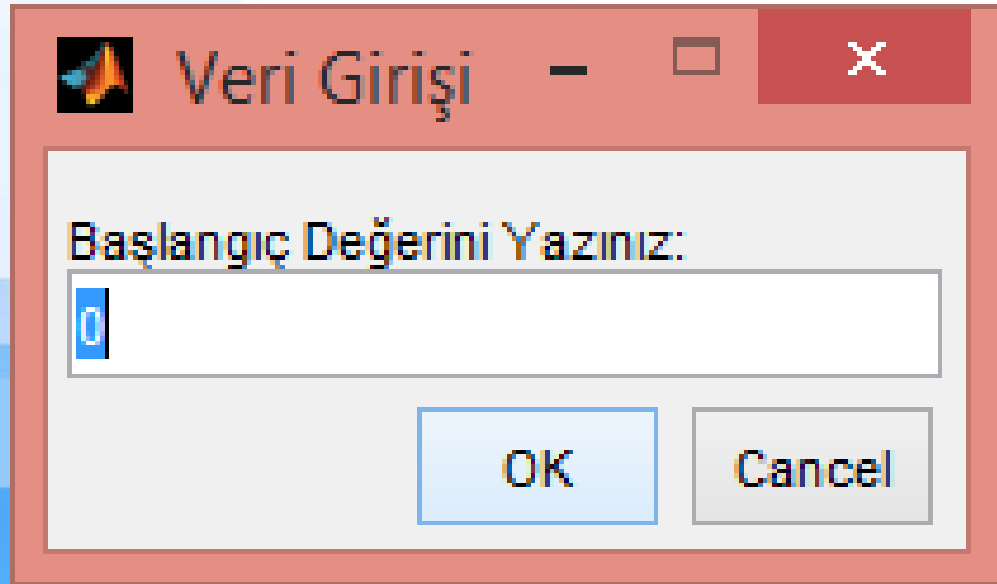
➤ `msgbox('Hatalı veri girildi!', 'Hata', 'error')`



Input Diyalog Penceresi ile Veri Girişi

- Input Diyalog penceresi ile veri girişi yapmak için şu komut kullanılır:
- `inputdlg(prompt,dlgtitle,dims,definput)`
- Örnek:
- `mesaj = {'Başlangıç Değerini Yazınız:'};`
- `baslik = 'Veri Girişi';`
- `boyut = [1 30];`
- `deger = {'0'};`
- `cevap = inputdlg(mesaj,baslik,boyut,deger)`

Input Diyalog Penceresi ile Veri Girişİ



Name ▲	Value	Size	Bytes	Class
abc baslik	'Veri Girişİ'	1x11	22	char
boyut	[1 30]	1x2	16	double
cevap	1x1 cell	1x1	116	cell
deger	1x1 cell	1x1	114	cell
mesaj	1x1 cell	1x1	166	cell

Input Diyalog Penceresi ile Veri Girişi

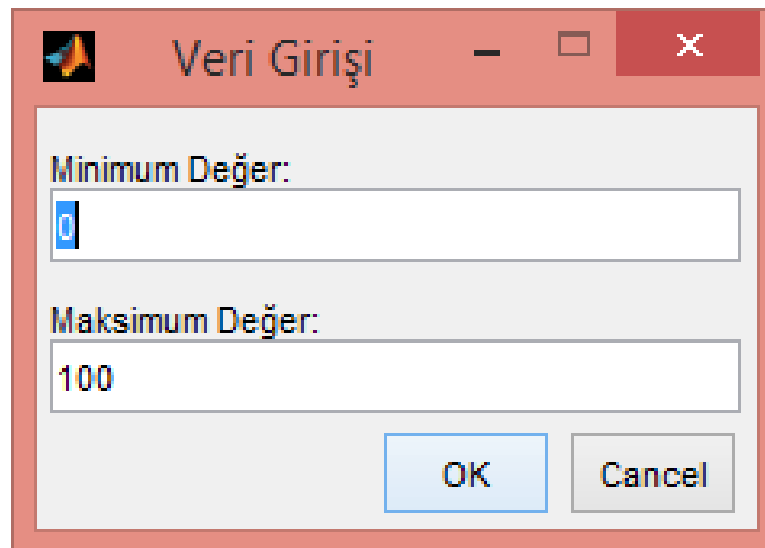
- Girilecek değer yazılıp "OK"a basılınca girilen veri **cell array** olarak saklanır:
- Cell Array olarak saklanan verinin sayısal veriye dönüştürülmesi için **str2num()** fonksiyonu kullanılır:
- `str2num(cevap{1})`

```
>> cevap
cevap =
    '20'

>> str2num(cevap{1})
ans =
    20
```

Birden Çok Veri Girişi Yapmak

- `mesaj = { 'Minimum Değer:', 'Maksimum Değer:' };`
- `baslik = 'Veri Girişi';`
- `boyut = [1 35];`
- `deger = {'0', '100'};`
- `cevap = inputdlg(mesaj, baslik, boyut, deger)`



Birden Çok Veri Girişi Yapmak

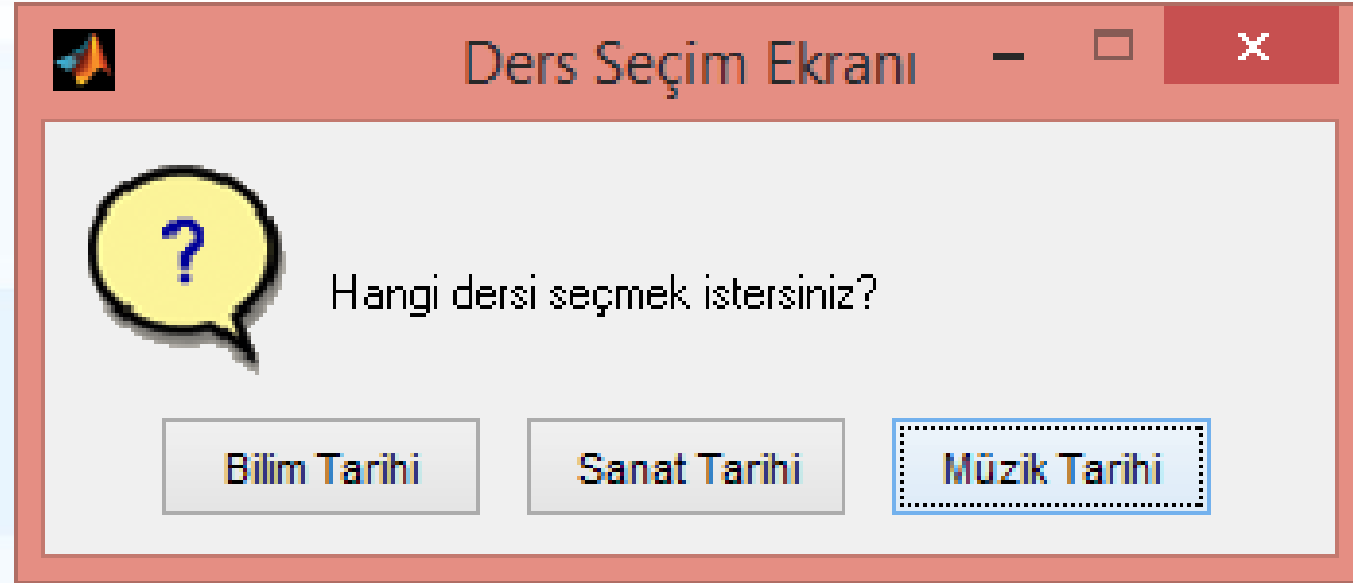
```
cevap =  
    '35'  
    '70'  
>> min=str2num(cevap{1})  
min =  
    35  
>> mak=str2num(cevap{2})  
mak =  
    70
```

questdlg ile Veri Girişi Yapılması

- Birden çok seçenek arasından bir tanesinin düğme üzerinde tıklanarak seçilmesi için kullanılır.
- `answer = questdlg(quest,dlgtitle,btn1,btn2,btn3,defbtn)`
- **Örnek:**
- `ders=questdlg('Hangi dersi seçmek istersiniz?','...
'Ders Seçim Ekranı','...
'Bilim Tarihi','Sanat Tarihi','Müzik Tarihi','Müzik Tarihi');`

questdlg ile Veri Giriş Yapılması

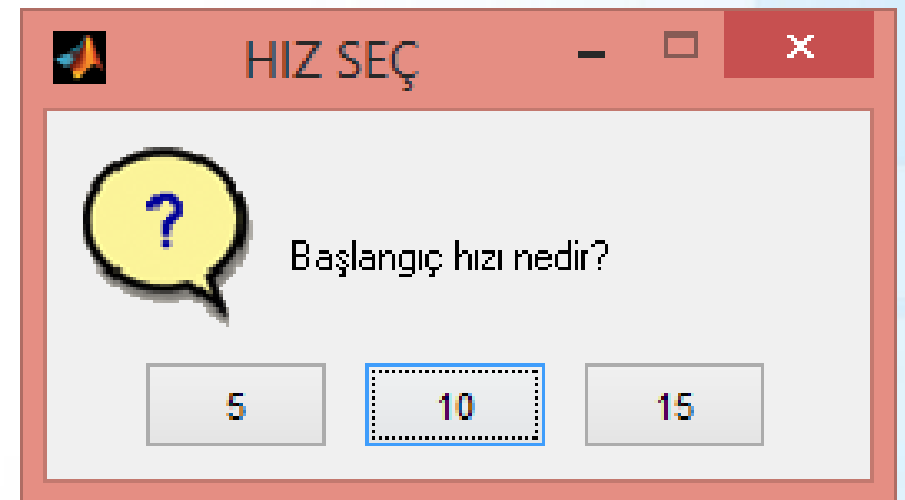
- Girilen veri **char** olarak tutulur.
- `defbtn` varsayılan olarak seçili düğmeyi göstermek için kullanılır.



questdlg ile Veri Girişi Yapılması

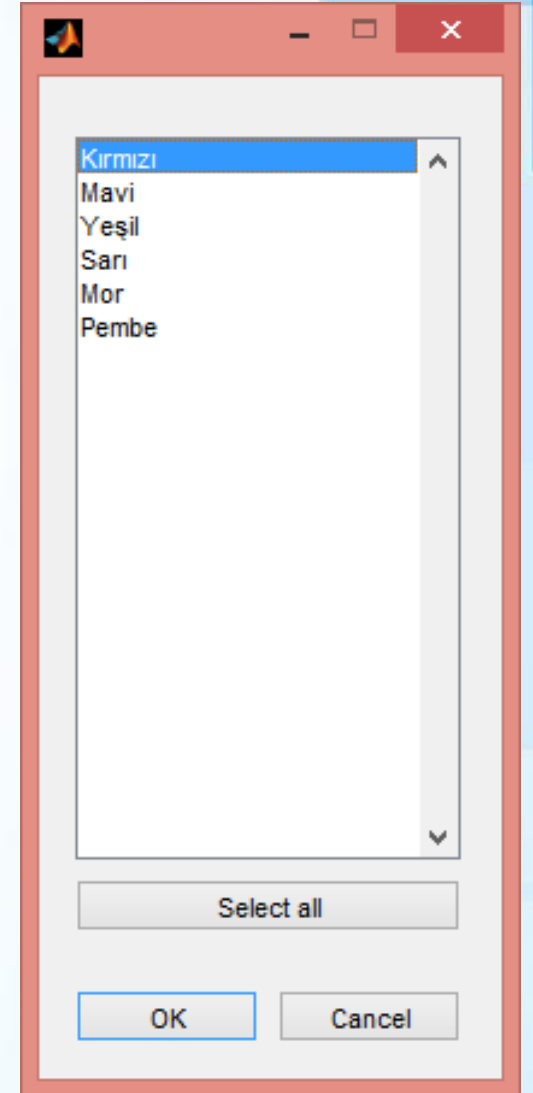
- Düğmelerdeki seçeneklere sayı yazılır ve questdlg ile bu sayılardan biri **char** olarak saklanabilir.
- Daha sonra bu değer `str2num()` fonksiyonu ile sayıya dönüştürülür.
- `hiz=questdlg('Başlangıç hızı nedir?','HIZ SEÇ','5','10','15','10');`
- `v=str2num(hiz);`

Workspace				
Name	Value	Size	Bytes	Class
hiz	'15'	1x2	4	char
v	15	1x1	8	double



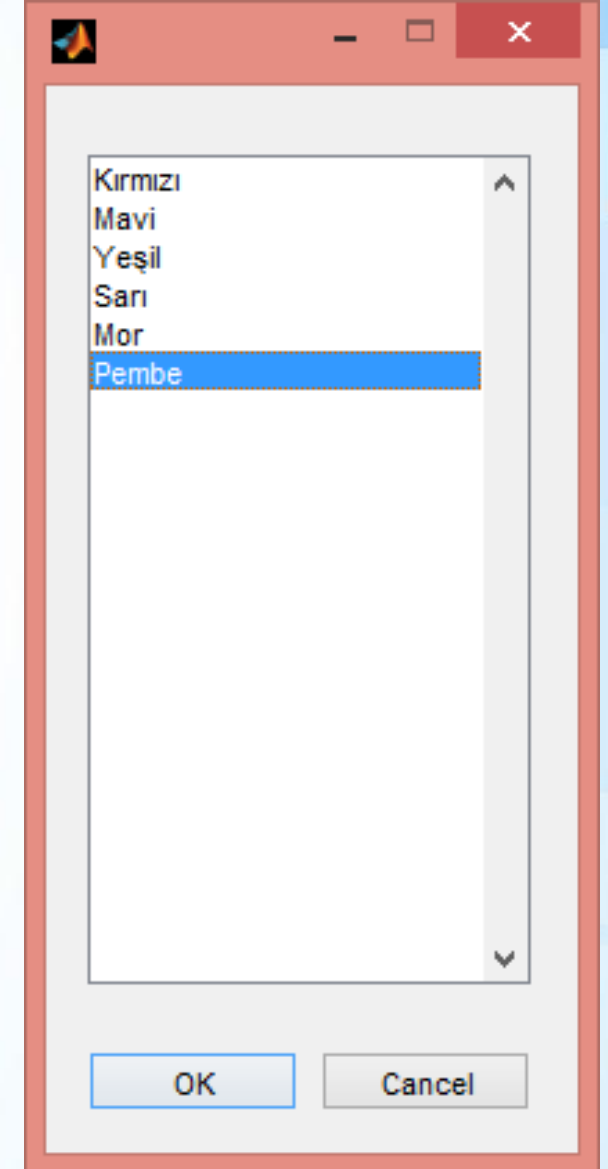
listdlg ile Liste İerisinden Seim

- Kullanıcının liste kullanarak seim yapması iin listdlg komutu kullanılır.
- Örnek:
- ```
renkler={'Kırmızı','Mavi','Yeşil','Sarı','Mor','Pembe'};
```
- ```
[indeks,tf] = listdlg('ListString',renkler);
```
- Seilen deėerin sırası sayı olarak indeks deėişkenine atanır.
- tf deėişkeni seimin yapılıp yapılmadığını kontrol etmek iin kullanılır. Kullanıcı renk seip "OK" düėmesine basarsa tf "1" deėerini, "Cancel" düėmesine basılması durumunda tf "0" deėerini alır.



listdlg ile Liste İçerisinden Seçim

- listdlg kutusunu sadece tek bir seçeneğin seçilmesine zorlamak için '**SelectionMode**','single' seçenekleri komuta eklenmelidir.
- Bu sayede "Select all" düğmesi ortadan kalkar.
- Sadece tek bir seçim yapılır.
- ```
[indeks,tf] =
listdlg('ListString',renkler,'SelectionMode','single');
```



# listdlg ile Liste İçerisinden Seçim

- Liste kutusunun başına mesaj eklemek için de **'PromptString',{'Mesaj'}** komutları eklenmelidir.
- `[indeks,tf] = listdlg('PromptString',{'Renk Seç'},'ListString',renkler,'SelectionMode','single');`

| Workspace |          |      |       |        |
|-----------|----------|------|-------|--------|
| Name ▲    | Value    | Size | Bytes | Class  |
| indeks    | 6        | 1x1  | 8     | double |
| renkler   | 1x6 cell | 1x6  | 728   | cell   |
| tf        | 1        | 1x1  | 8     | double |

