

Összehasonlító operátorok

- $<$, \leq , $>$, \geq
összehasonlítás, a szokásos jelentésekkel
- $==$
az egyenlőség tesztelése
- \sim
„nem egyenlő”

Két mátrixra is alkalmazhatóak, ilyenkor elemenként történik az összehasonlítás

A visszaadott érték minden esetben egy logikai változó:

`logical 1` (igaz)

`logical 0` (hamis)

Példa.

```
>> 1<2
```

```
ans =
```

```
logical
```

```
1
```

```
>> 1>2
```

```
ans =
```

```
logical
```

```
0
```

```
>> [1,2,3]<[-1,1,4]
```

```
ans =
```

```
1×3 logical array
```

```
0    0    1
```

Logikai operátorok

- $A \& B$ A és B
- $A \& \& B$ A és B (rövid kiértékelés)
- $A | B$ A vagy B
- $A || B$ A vagy B (rövid kiértékelés)
- $\sim A$ nem A
- $\text{xor}(A, B)$ A kizáró vagy B

A kiértékelések elemenként történnek.

rövid kiértékelés: ha az első kifejezés értéke egyértelműen eldönti a teljes kifejezés értékét, akkor a második kifejezést nem értékeli ki.

Logikai függvények

- `all(A)`
egy sorvektorral tér vissza, oszloponként megvizsgálja, hogy minden elem 0-tól különböző-e (ugyanaz, mint `all(A,1)`)
- `all(A,2)`
egy oszlopvektorral tér vissza, soronként megvizsgálja, hogy minden elem 0-tól különböző-e
- `any(A)`
egy sorvektorral tér vissza, oszloponként megvizsgálja, hogy van-e 0-tól különböző elem (ugyanaz, mint `any(A,1)`)
- `any(A,2)`
egy oszlopvektorral tér vissza, soronként megvizsgálja, hogy van-e 0-tól különböző elem

find

- `ind=find(A)`
a nemnulla elemek sorszámaival tér vissza, ahol az elemek számozása oszlopfolytonosan történik. Ha A egy sorvektor, akkor a visszaadott érték is az, egyébként oszlopvektor
- `ind=find(A,n)`
az A első n darab nemnulla elemének sorszámával tér vissza
- `ind=find(A,n,last)`
az A utolsó n darab nemnulla elemének sorszámával tér vissza
- `[rowI,colI]=find(A)`
a nemnulla elemek sor- és oszlopindexeivel tér vissza
- `[rowI,colI,elem]=find(A)`
a nemnulla elemek sor- és oszlopindexeivel, illetve a nemnulla elemekkel tér vissza

Keresés tömbökben

Legyen $a=[1,-2,-1,0,5,4]$

- Számoljuk össze hány negatív eleme van a -nak!

Az $a < 0$ kifejezés értéke egy logikai vektor: 1, ha a feltétel teljesül az adott elemre, 0, ha nem:

```
ans =
```

```
1×6 logical array
```

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

Ha ennek a vektornak a koordinátáit összeadjuk, éppen a negatív elemek számát kapjuk, azaz **az a vektor negatív elemeinek száma:**

```
>> sum(a<0)
```

```
ans =
```

```
2
```

Keresés tömbökben

Legyen $a=[1,-2,-1,0,5,4]$

- Soroljuk fel az a negatív elemeit!

Ha az $a < 0$ logikai vektort beírjuk az a vektor indexargumentumába, akkor pontosan az 1-eseknek megfelelő helyen álló elemeket, azaz az a vektor negatív elemeit sorolja fel:

```
>> a(a<0)
```

```
ans =
```

```
    -2    -1
```

- Adjuk össze az a negatív elemeit!

Az előbb eredményül kapott vektor elemeit kell csak összeadnunk:

```
>> sum(a(a<0))
```

```
ans =
```

```
    -3
```

Legyen

$$A = \begin{bmatrix} 1 & -2 & -1 \\ 0 & 5 & 4 \end{bmatrix}$$

- Számoljuk össze hány negatív eleme van A-nak!

Az $A < 0$ kifejezés értéke:

```
ans =
```

```
2×3 logical array
```

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

$\text{sum}(A < 0)$ vagy $\text{sum}(A < 0, 1)$ adja a negatív elemek számát
oszloponként:

```
>> sum(A < 0)
```

```
ans =
```

```
0    1    1
```


`sum(A<0,2)` adja a **negatív elemek számát soronként**:

```
>> sum(A<0,2)
```

```
ans =
```

```
2
```

```
0
```

Az összes negatív elem száma:

```
>> sum(A<0,'all')
```

```
ans =
```

```
2
```

Legyen

$$A = \begin{bmatrix} 1 & -2 & -1 \\ 0 & 5 & 4 \end{bmatrix}$$

- Soroljuk fel az A negatív elemeit!

```
>> A(A<0)
```

```
ans =
```

```
-2
```

```
-1
```

(Az elemeket mindig egy oszlopvektorban, oszlopfolytonosan sorolja fel.)

- Adjuk össze az A negatív elemeit!

Az előbb eredményül kapott vektor elemeit kell csak összeadnunk:

```
>> sum(A(A<0))
```

```
ans =
```

```
-3
```

Legyen $a=[1,-2,-1,0,5,4]$

- Számoljuk össze hány -1 és 2 közé eső eleme van a -nak! (A határok nélkül.)

A $(a>-1)\&(a<2)$ kifejezés értéke egy logikai tömb: 1, ha egy elemre mindkét feltétel teljesül, egyébként 0:

```
ans =  
1×6 logical array
```

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

A megfelelő elemek számát megkapjuk, ha ezen utóbbi tömb elemeit összeadjuk:

```
>> sum((a>-1)&(a<2))
```

```
ans =
```

```
2
```