Introduction to MATLAB

Week 3 Lecture 1

Polina Cherepanova
cherepanova@princeton.edu

Relational operators

- equal
- not equal
- greater than
- less that
- greater than or equal to
- less than or equal to

```
%% logical variables
true;
false;

number=10;
number==10
number~=10
number≥3
number≤3
```

```
>> %% find the index of a certian element
array=[23 45 -3 -7 -10 30 5 8];

array==5

ans =

  1×8 logical array

   0   0   0   0   0   0   1   0
```

Find index of specific elements in MATLAB

Use the built-in function ‘find’

Find function will directly give the index at which the condition is true

```
>> array=[23 45 -3 -7 -10 30 5 8];
>> %% use find function (will direclty give you the index position)
index_5=find(array==5)

index_5 =

     7
```

```
>> %% find the index of all the numbers greater than a number
array=[23 45 -3 -7 -10 30 5 8];
index_more_than_5=find(array>5)

index_more_than_5 =

     1     2     6     8

>> index_more_than_and_equal_5=find(array>=5)

index_more_than_and_equal_5 =

     1     2     6     7     8
```

```
>> %% replace numbers greater than a specific number by another number
%%% let us replace all the numbers greater than 6 by 20
array=[23 45 -3 -7 -10 30 5 8];
index_more_than_6=find(array>6)
array(index_more_than_6)=20

index_more_than_6 =

     1     2     6     8


array =

    20    20    -3    -7   -10    20     5    20
```

```
%% find index of all zero elements
arrays_with_zeros=[23 0 0 45 -3 -7 0 -10 0 5 8 0];
zero_indices=find(arrays_with_zeros==0)

%%% replace the zero elements by another number e.g. 100
arrays_with_zeros(zero_indices)=100
```

```
%% find NaN elements
arrays_with_nan=[23 NaN NaN 45 -3 -7 NaN -10 NaN 5 8 0];
isnan(arrays_with_nan)
index_nan=find(isnan(arrays_with_nan))
```

```
%% find non-NaN elements
arrays_with_nan=[23 NaN NaN 45 -3 -7 NaN -10 NaN 5 8 0];
~isnan(arrays_with_nan)
index_nan=find(~isnan(arrays_with_nan))
```

```
%% replace all NaN by -100
arrays_with_nan(index_nan)=-100
```

Logical operators

- and
- or
- not
- all
- any

‘or’ and ‘and’

```
%% using 'or' function
array_new=[2 5 -2 1 7 5 7 4 -3 6];
index_or=find(array_new>3 | array_new<0)
```

```
%% using 'and' function
array_new=[2 5 -2 1 7 5 7 4 -3 6];
index_and=find(array_new>3 & array_new<7)
```

```
%% finding if a matrix is empty
empty_matrix=[];
not_empty_matrix=[1 2 3];
isempty(empty_matrix)
isempty(not_empty_matrix)
```

‘not’

```
vec = [2 5 10 -4 0 2.5 7 -9];

~isempty(vec)
not(isempty(vec))


find(vec <= 0)
find(~(vec <= 0))
find(not(vec <= 0))
```

‘any’ and ‘all’

```
vec1 = [1 2 3 4 5];
any(vec1 >= 4)
all(vec1 >= 4)
```

```
mat1 = [1:5; 6:10; 11:15];
any(mat1 < 4)
any(mat1 < 12, 'all')
```