EGR 111
Relational Operators

This lab introduces relational operators and logical operators which allows MATLAB to compare values and count the number of values that satisfy a given condition.

New MATLAB Commands: <, >, ==, <=, >=, ~=, &, |, ~

1. Relational Operators

The relational operators in Table 1 are used to compare two numerical values.

Table 1. Relational Operators

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
~=	Not equal to

Let's look at some simple examples. First, determine if 5 is less than 8:

```
>> 5 < 8
ans =
     1
```

Notice that the answer is 1. In MATLAB, 1 means *true* and 0 means *false* (Actually, any non-zero value also is treated as true.) That is, it is *true* that 5 is less than 8.

Next let's see if 5 equals 8:

```
>> 5 == 8
ans =
     0
```

Note that the double equal sign == is used when we are comparing two values (5 == 8), whereas the single equal sign = is used when assigning a value to a variable (x = 4).

Relational operators also work on variables, vectors, and matrices. For example, if we want to identify which values in vector x are negative, we could compare x to 0 as follows.

```
>> x = [-20 -10 10 20 30]
x =
   -20   -10    10    20    30
>> x < 0
ans =
     1     1     0     0     0
```

The command `x < 0` above compares each element of x to 0, and returns a vector of ones and zeros where the ones indicate for which elements of x the comparison `x < 0` is true.

When using these operators, one has to be careful of operator precedence. The arithmetic operators have precedence over the relational ones, so in the following command, the addition and division are computed first resulting in 7 < 8, and then the comparison is performed.

```
>> 3 + 4 < 16 / 2             % + and / are executed first
ans =
     1
```

2. Logical Operators

Suppose you wanted to see if the variable x is within a given range, say between 1 and 10 (that is to say that x ≥ 1 and x ≤ 10). For this type of operation, we need the following logical operators.

Table 2. Logical Operators

Operator	Description
&	AND. A binary operator which takes two operands and yields 1 if both operands are 1, and 0 otherwise.
|	OR. A binary operator which takes two operands and yields 1 if at least one of the operands is 1, and 0 otherwise.
~	NOT. A unary operator which takes one operand and yields the negation. If the operand is 1, it yields 0. If the operand is 0, it yields 1.

The AND operator is 1 if the first operand is *true* AND the second operand is *true* (that is both operands are 1).

```
>> 1 & 1
ans =
     1
```

The OR operator is 1 if the first operand is *true* OR the second operand is *true* (that is if either operand is 1 or both operands are 1).

```
>> 1 | 0
ans =
     1
```

The NOT operator is *true* if the operand is NOT *true* (that is if the operand is 0).

```
>> ~0
ans =
     1
```

Try out the other inputs to the AND, OR, and NOT operators to verify that they work the way you think they do.

To test to see if x is between 1 and 10 (that is to say that x ≥ 1 and x ≤ 10), we would write the following (choosing x = 4 as an arbitrary example value):

```
>> x = 4
x =
     4
>> x >= 1 & x <= 10
ans =
     1
```

Try out other values for x to see if they work as expected.

Just like with relational operators, one must be careful with precedence. In the above command, the relational operators (>= and <=) are computed first, followed by the logical operator (AND) (see Table 3).

Table 3. Operator Precedence

Precedence	Operation
1	Parentheses
2	Exponentiation (^)
3	Logical NOT (~)
4	Multiplication (*) and division (\ )
5	Addition (+) and subtraction (-)
6	Relational operators (> < >= <= == ~=)
7	Logical AND (&)
8	Logical OR (|)

If two operators have the same precedence, the expression is executed left to right.

Exercise 1: Evaluate each of the following expressions by hand first and then use MATLAB to check your answer.

```
>> x = -2; y = 5;
>> -5 < x & x < -1

  ans = _____________

>> ~(y < 7)

  ans = _____________

>> ~y < 7

  ans = _____________

>> ~(y >= 8) | (x < -1)

  ans = _____________
```

3. Logical Functions

MATLAB has other built-in functions that can aid in determining logical relationships among values (see Table 4). Try the following functions on some sample inputs to see how they work.

Table 4. Logical Functions

Function	Description
`any(A)`	If A is a vector, returns 1 if any of the elements A is *true* (non-zero) and 0 otherwise.
`all(A)`	If A is a vector, returns 1 if all of the elements in A are *true* (non-zero) and 0 otherwise.
`find(A)`	If A is a vector, returns the indices of the elements that are *true* (non-zero).
`sum(A)`	If A is a vector, returns the sum of the values in the vector.

For example, in order to count the number of negative elements in the vector x, we could use the following commands:

```
>> x = [-20 -10 10 20 30]
x =
   -20   -10    10    20    30
>> sum(x < 0)
ans =
     2
```

The `x < 0` command above results in the vector [1 1 0 0 0] because the comparison x < 0 is true for the first two elements. Then the command `sum(x < 0)` adds up the elements: 1 + 1 + 0 + 0 + 0 = 2.

Exercise 2. Download the file "Portland Weather Data 2013.xlsx" from the course website, save it to your P:\MATLAB folder, and open it using Excel. Note that the first column is the date and the second column is Portland's maximum daily temperature in degrees Fahrenheit. Load the Excel file into MATLAB by typing the following command:

```
[data,txt,raw] = xlsread('Portland Weather Data 2013.xlsx')
```

After the above command, the numerical data from the Excel file will be in the MATLAB variable data. Since the first column in the Excel file contains strings that represent dates, that column will not be stored in the variable data. Therefore the first column of data represents the maximum temperature in degrees Fahrenheit.

Plot the maximum temperature data, label the axes, and title the plot.

Use MATLAB to find the number of "cold" days in Portland in 2013 if we define a "cold" day as one where the maximum temperature is 40 °F or below (temperature ≤ 40 °F).

Also use MATLAB to find the number of "nice" days in Portland in 2013 if we define a "nice" day as one where the maximum temperature is between 70 and 85 °F inclusive (70 ≤ temperature ≤ 85 °F).

Checkpoint 1: Show the instructor your commands and results from Exercise 2.