Concepts for Today's Class:
- steps in problem solving
- MATLAB syntax: underlying structure
- syntax versus semantics

Reading:
- See course web site for details, we'll cover material in:
- pgs 6-40   (in class)
- pgs 3-6, 75-80, 89-101   (labs)
- pgs 557-564   (general reference, will show up in classes)
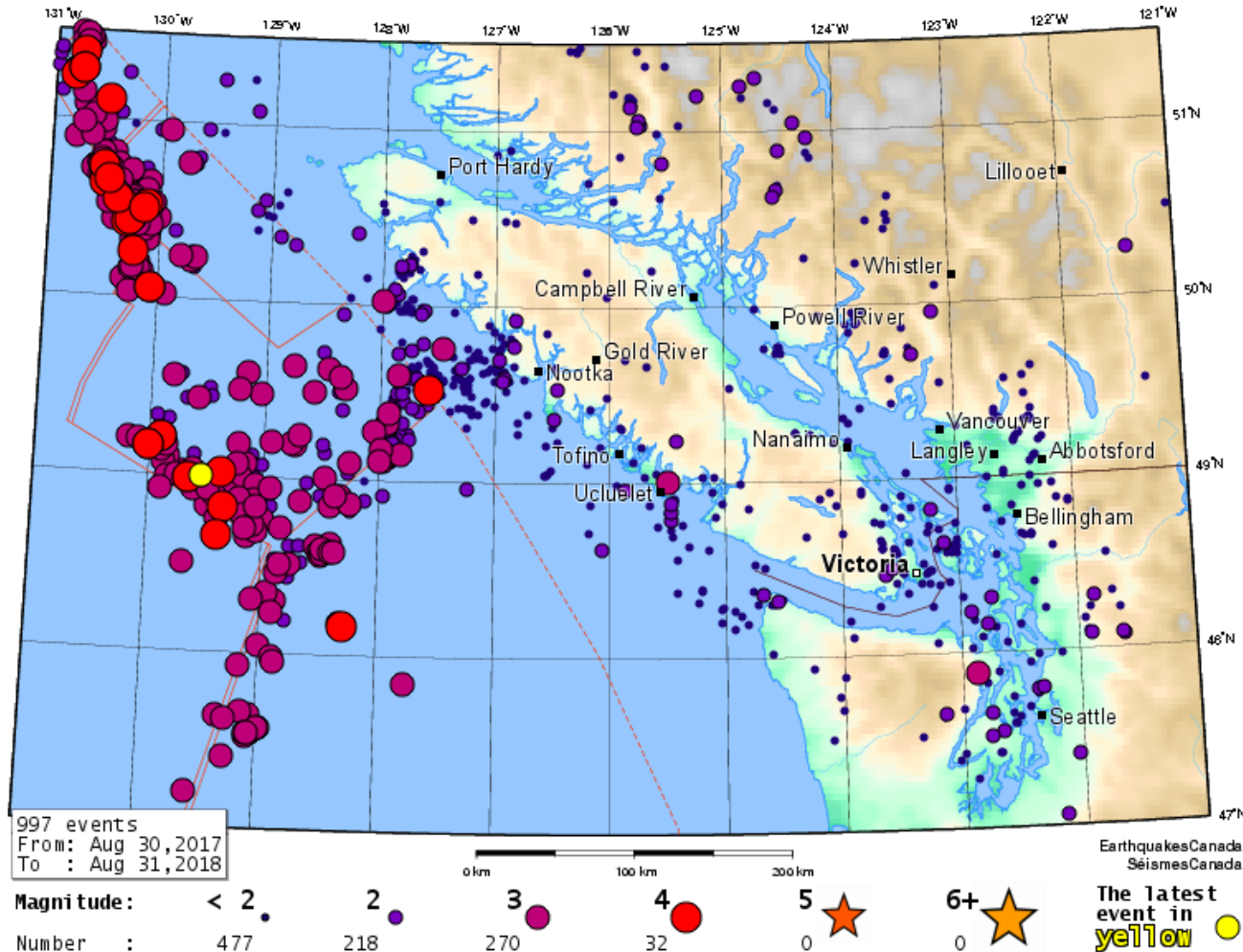
Lab:

- bring text
- Labs due at **4pm on Fridays**

TA hours, EOS-Main 203 – make sure to note which weeks these carefully!

Friday (wks 3,4,5,6,8,9,10,11):   9:30 am – 10:30 am
Monday (wks 7,10 **only**):   11:00 am – noon.

Good Problem Solving Techniques

Work in pairs and write down the steps in algorithm design (problem solving using MATLAB) mentioned in the text book (or if you didn't get that far in the reading just write down what you think are reasonable steps)

Good Problem Solving Techniques

We will adopt the 5-step program **(NOTE: this is more complete than the approach in the text book on page 76)**
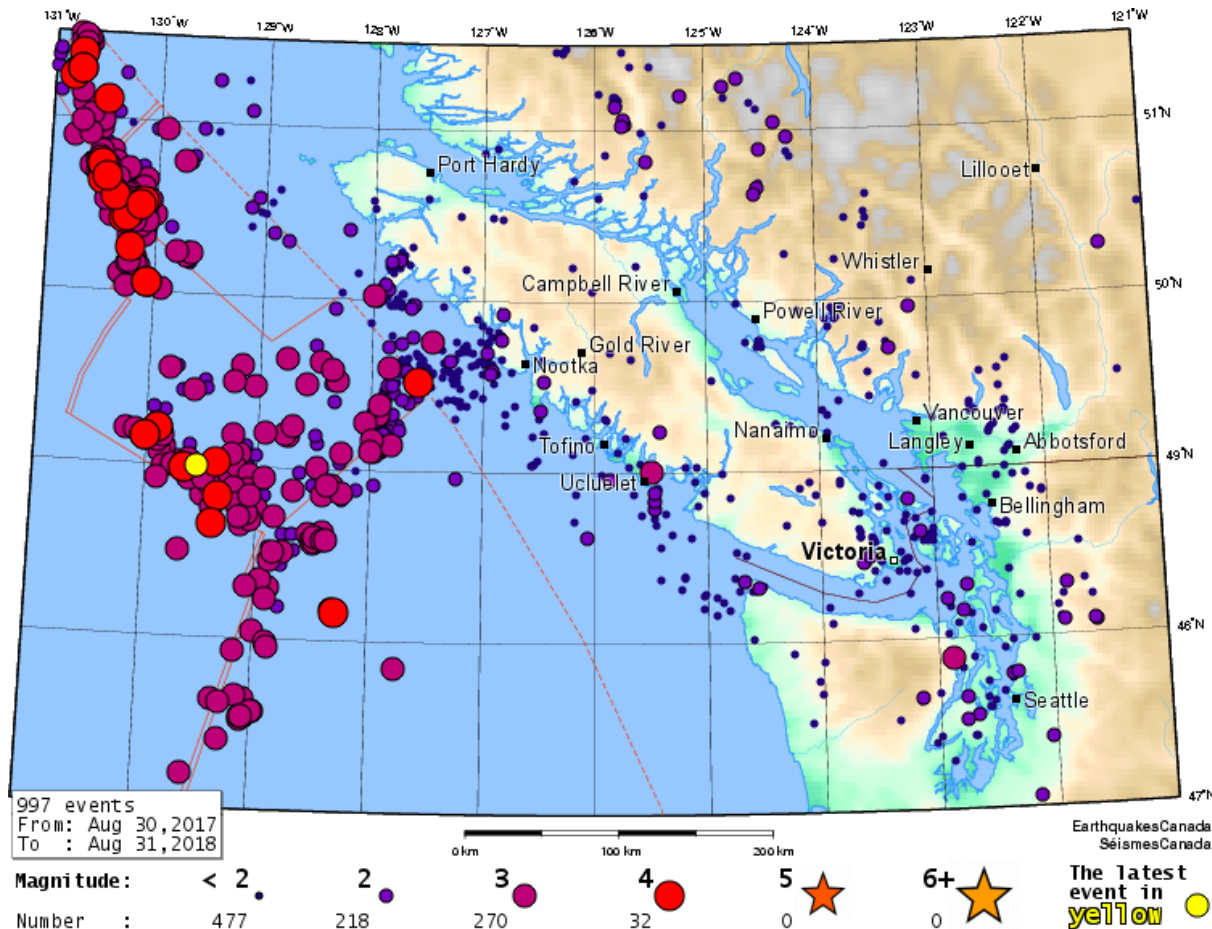
1. State the problem clearly

2. Describe your input and output info / data

3. Write down the problem-solving procedure IN WORDS
    - pay special attention to the logic
    - if possible work the problem "by hand" for a simple set of data

4. Develop a MATLAB code to do (3)

5. Test your code with a variety of data or different cases

Example: What was the largest earthquake in the Vancouver / Vancouver Island region in the past year?

http://www.iris.edu/hq/audience/public/earthquakes
http://earthquake.usgs.gov/regional/neic/
http://earthquakescanada.nrcan.gc.ca/index-eng.php

997 events
From: Aug 30, 2017
To : Aug 31, 2018

Magnitude:	< 2	2	3	4	5	6+	The latest event in **yellow**
Number :	477	218	270	32	0	0	

EarthquakesCanada
SéismesCanada

Goal: find the largest earthquake in the Pacific Northwest during 08/30/2017–08/31/2018. I have only printed out eqs w/ magnitudes larger than 4.0

1. Write down input and output in words

2. Give numeric value for the actual output

3. Write down the procedure you used to get your answer

Finding the Largest Earthquake

My procedure or algorithm:

1.	read magnitude on first line, and remember

2.	read magnitude on second line

3.	compare with first, and

4.	remember only the largest of the 2

5.	read magnitude on the next line

6.	compare with largest, and

7.	remember the largest of these two

8.	repeat steps 5-7, until reach end of list

9.	tell you my largest magnitude

Finding the Largest Earthquake

My procedure or algorithm:

read mag1
read mag2

Is mag2 > mag1 ?
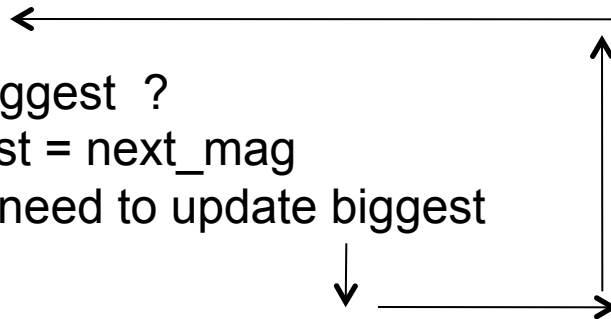   YES: biggest = mag2
    NO: biggest = mag1

read next_mag
Is next_mag > biggest ?
    YES: biggest = next_mag
    NO: don't need to update biggest

write out biggest

- how do we turn this into matlab code?

MATLAB: Programming elements (overview)

The structure of MATLAB

1. Lexical Elements
        - building blocks

Programming language
names, operators, special characters

Human Language
words, symbols, punctuation

2. Syntax
        - set of rules to combine lexical elements into legitimate constructs

Programming language
how to build expressions, statements,
functions & programs

Human Language
how to build sentences & paragraphs

Lexical Elements (Building Blocks)

1. Names
   a) Variable names: store data: e.g., our user-defined variable: mag1
   b) Reserved words: have a special meaning, cannot be redefined; e.g. for
   c) Function names: sequences of code; e.g. sin

2. Operators
   a) Logical: e.g., &&   text p. 20
   b) Arithmetic: e.g., +, -, *, /   text p. 12-13
   c) Relational: e.g., >, >=, <, <=   text p. 19

3. Special characters: keyboard characters with special meaning in MATLAB.
   e.g., ; {} ()

Lexical Elements (Building Blocks)

1. Names: variable names, reserved words, function names

2. Operators: logical, arithmetic, relational

3. Special characters: keyboard characters with special meaning in MATLAB.

```
mag1 =4.9;
mag2 =5.3;
if (mag1 > mag2)
        max_mag = mag1;
else
        max_mag = mag2;
end
```

*Worksheet
Part D*

Lexical Elements (Building Blocks)

How are variables assigned in MATLAB?

`>> mag1 = 4.9`

What does this do?
1. Creates a space in memory
2. Gives the space the name "mag1". We call mag1 a VARIABLE
3. Stores the value 4.9 in that space (variable)
4. Echoes back this information
5. The prompt reappears

`>> mag1 = 3.7`   **overwrites "4.9" with "3.7"**

Note
- "=" does not mean = in the mathematical sense. Instead, it means "assign the number on the RHS to the variable on the LHS"

- We can do math on the RHS, mixing numbers and variables

Syntax

Lexical elements are combined using the set of rules known as "syntax"

> I cdnuolt blveiee that I cluod aulacity uesdnatnrd what I was rdgnieg.

⇒Important difference between human and programming languages is tolerance to syntax errors versus tolerance to spelling/grammar errors

Examples of Syntax Errors

Usually / often your code will not run if you make a syntax error:

```
>> x=10;                              defined my variable x and assigned it a value of 10


>> y=cos(x                            trying to compute cos(x) and assign it to variable y
??? y=cos(x                           Correct: y = cos(x)
           |
Error: Expression or statement is incorrect--possibly unbalanced (, {, or [.



>> y=2x                               trying to compute "2 times x" and assign it to "y"
??? y=2x                              Correct: y = 2*x (spaces around = don't matter)
       |
Error: Unexpected MATLAB expression.
```

Syntax versus Semantics

When a program doesn't work this can happen because:

a)   SYNTAX is incorrect - e.g.,

>> distance = speed *

- multiplication (*) needs numbers or variables on both sides of *

b)   SEMANTICS are incorrect

>> distance = speed / time

- may or may not run but will <u>certainly</u> give you the wrong answer!!
The program you wrote is not the program you meant to write!

can reduce both types of errors using
good problem solving techniques

Example of Semantics Error

Here's a code snippet for finding the largest of 2 earthquakes.
What is wrong?

```
mag1 =4.9;
mag2 =5.3;
if (mag1 < mag2)
        max_mag = mag1;
else
        max_mag = mag2;
end
```

Wrap Up

Steps in problem solving

      1. state problem clearly
      2. define the input/output
      3. write down algorithm by hand: ie., think the problem through
      4. code in MATLAB
      5. test, test, test…

Underlying structure of MATLAB syntax
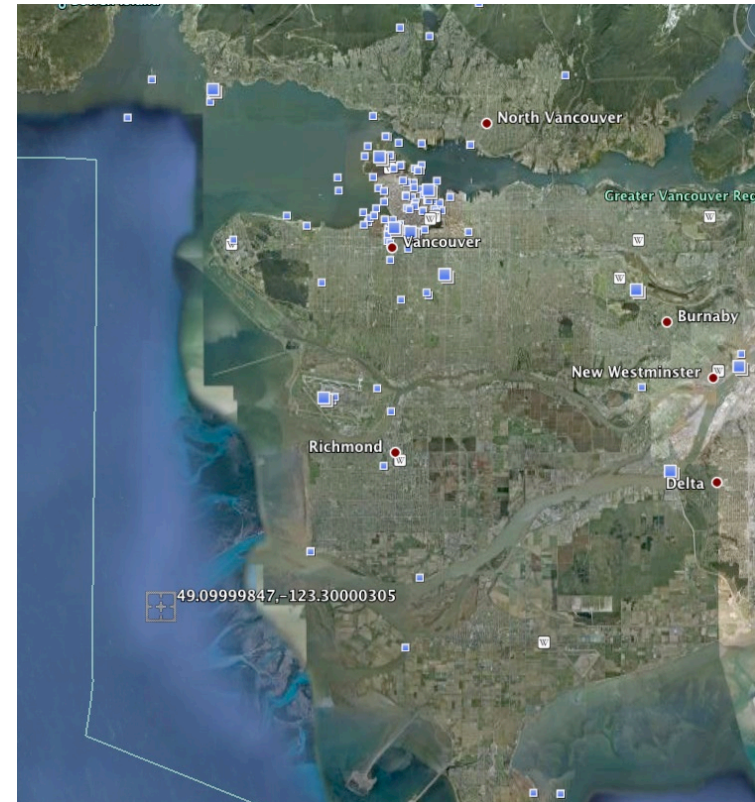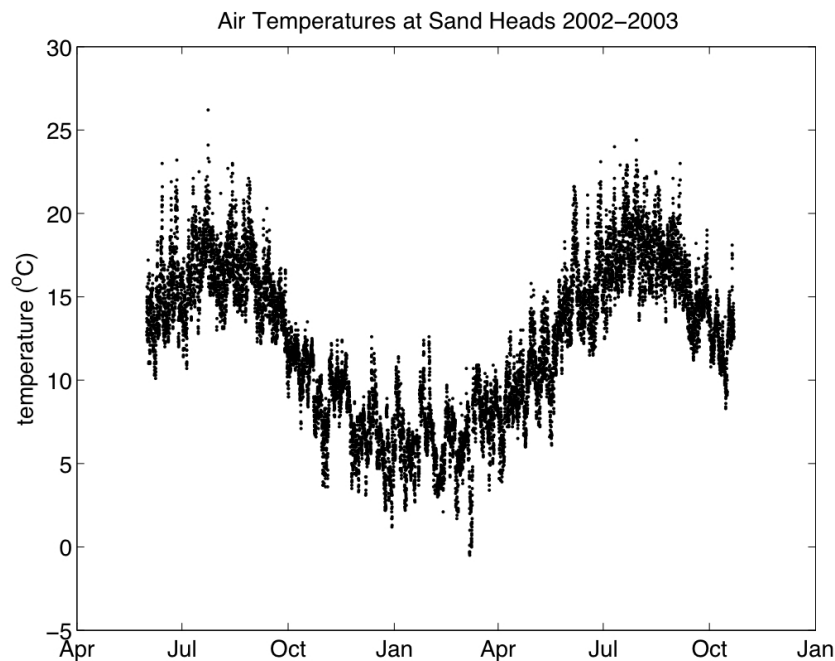
      - building blocks: names, operators, special characters

Syntax versus semantics

      - syntax errors are about bugs in step 4 above
      - semantics errors usually result from skipping step 3

Lab Overview

•become familiar with MATLAB environment
•load, plot and save data
•try some simple operations
•write MATLAB scripts

Air Temperatures at Sand Heads 2002–2003

a) pair programming
b) need either an account, or your own version of MATLAB
c) turn in requested file on Canvas (not graded but will use later in term)