

Announcements

- Lab 5 on Wednesday
- A2 resubmissions
- Upcoming lecture topics:
 - Image processing
 - Function handles

Agenda

- Review file I/O
- Vectorized filtering
- Permutations & sorting
- Searching

Read lines into cell array

1. Open file
 - `fopen()`
2. Read it line-by-line until end-of-file
 - `fgetl()`, `feof()`
3. Close file
 - `fclose()`

Closing a file is like the `end` keyword – need to tell MATLAB when you're done

1 & 3: Open (and close) file

```
fid = fopen('statePop.txt', 'r');
```

An opened file has a file ID, here stored in variable **fid**

Name of the file opened. **txt** and **dat** are common file name extensions for plain text files

'**r**' indicates that the file has been opened for reading

Built-in function to open a file

```
fclose(fid);
```

; because file commands return status codes

2: Read each line and store it in cell array

```
fid = fopen('statePop.txt', 'r');
```

```
k= 0;
```

```
while ~feof(fid)
```

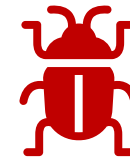
```
    k= k+1;
```

```
    Z{k}= fgetl(fid);
```

```
end
```

```
fclose(fid);
```

*False until end-of-file
is reached*



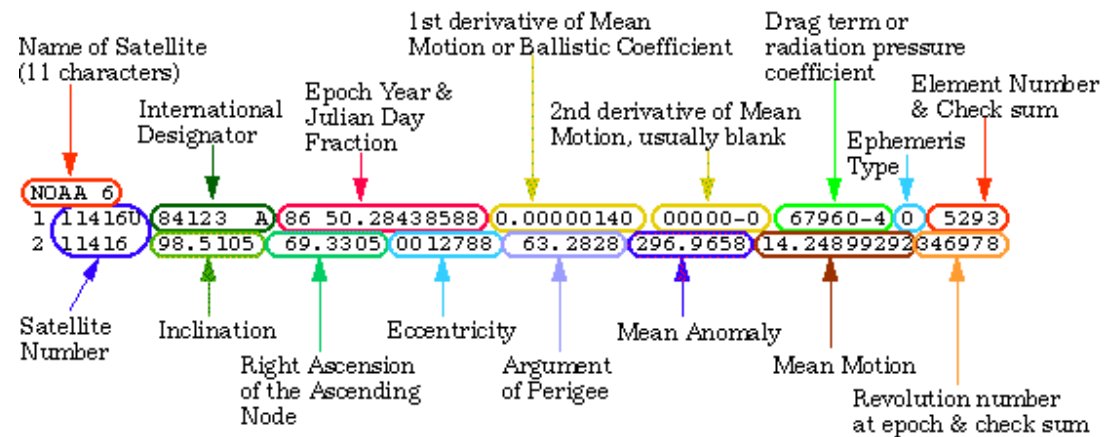
Doesn't work for
empty files

*Get the next line.
(Each call gets one line; you cannot
make it skip lines or go to a
specific line.)*

Storing only a selected portion from a big file

- If only interested in a specific piece of data in the file, storing everything may be wasteful
- If random access is not required, can process each line as it is read

Example: NORAD two-line elements



ISS (ZARYA)

```
1 25544U 98067A 19280.43177083 .00000288 00000-0 13040-4 0 9993
2 25544 51.6437 164.6585 0007556 123.5429 237.5675 15.50172544192676
:
```

STARLINK-74

```
1 44293U 19029BL 19280.46307273 .00000774 00000-0 72445-4 0 9999
2 44293 53.0058 280.3384 0001435 93.2755 266.8397 15.05496611 21751
```

STARLINK-53

```
1 44294U 19029BM 19279.64653505 .00000628 00000-0 62400-4 0 9998
2 44294 52.9988 283.1290 0000873 99.6752 260.4335 15.05478127 19808
```

COSMOS 2534 [GLONASS-M]

```
1 44299U 19030A 19279.63973935 .00000042 00000-0 00000+0 0 9999
2 44299 64.7328 275.7191 0015277 282.8642 34.0841 2.13101948 2816
```

Website example: satellite launch year

1. Read **line** (satellite name)
2. While name is not ISS
 1. Read 2 lines (skip)
 2. Read line (satellite name)
3. Read line (record 1)
4. Extract characters **10 & 11**
5. Convert to number (`str2double`), interpret as year

SCD 2

```
1 25504U 98060A 19288.18395014 .00000230 00000-0 13957-4 0 9992
```

```
2 25504 24.9967 317.5526 0017113 331.0386 103.7958 14.44077629107938
```

ISS (ZARYA)

```
1 25544U 98067A 19280.43177083 .00000288 00000-0 13040-4 0 9993
```

```
2 25544 51.6437 164.6585 0007556 123.5429 237.5675 15.50172544192676
```

:

STARLINK-53

```
1 44294U 19029BM 19279.64653505 .00000628 00000-0 62400-4 0 9998
```

```
2 44294 52.9988 283.1290 0000873 99.6752 260.4335 15.05478127 19808
```

COSMOS 2534 [GLONASS-M]

```
1 44299U 19030A 19279.63973935 .00000042 00000-0 00000+0 0 9999
```

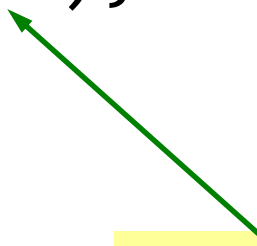
```
2 44299 64.7328 275.7191 0015277 282.8642 34.0841 2.13101948 2816
```

Writing a file

```
fid = fopen('geneData.txt', 'w');
```

```
fprintf(fid, 'x = %.17g', x);
```

```
fclose(fid);
```



'w' indicates that the file is to be opened for writing

Use 'a' for appending

Writing cell arrays to files

```
function cellArray2file(CA, fname)
% Write each string in cell array
% CA as a line in a file.
fid= fopen(fname, 'w');
for k= 1:length(CA)
    fprintf(fid, '%s\n', CA{k});
end
fclose(fid);
```

Recommendation

Read files into cell arrays when:

1. File is line-oriented text
2. File contents fit in memory
3. Processing involves jumping between lines

Vectorization: separate “what” from “how”

```
function [a,b] = f(x,y)
n= length(x);
a= zeros(1,n);
b= inf;
for k= 1:n
    a(k)= sqrt(x(k)^2 + ...
              y(k)^2);
    if a(k) < b
        b= a(k);
    end
end
```

```
function [a,b] = f(x,y)
a= sqrt(x.^2 + y.^2);
b= min(a);
```

Goal: Minimize mental energy (not necessarily lines of code)

Vectorized logic with char arrays

Hamming distance

How many letters must change to transform one word into another?

in other words,

How many letters are different in two words of the same length?

- `str1 = 'hello';`
`str2 = 'jello';`
- `str1 ~= str2`
 - `[1 0 0 0 0]`
- `sum(str1 ~= str2)`
 - `1`

Logical indexing of char arrays

Play *Wheel-of-Fortune* with two one-line functions

```
function s = makePuzzle(s)
% Replace all letters in s with
% underscores (leaving spaces)
s(isletter(s)) = '_';
```

```
function b = guessLetter(c,b,s)
% Replace characters in b with c
% wherever s has character c
b(s==c) = c;
```

```
b= makePuzzle(secret)
' _____ '
```

```
b= guessLetter('i',b,secret)
'_i_____ i_ ___i_'
```

Arrays: What we didn't tell you

```
m= [ 2  3  5  7;  
    -2 1  0  7;  
     5  2 -1  8 ]
```

```
L= m>3  
   [ 0  0  1  1;  
     0  0  0  1;  
     1  0  0  1 ]
```

```
P= m>3 | m<0  
   [ 0  0  1  1;  
     1  0  0  1;  
     1  0  1  1 ]
```

```
a= m(m>3)  
   [ 5; 5; 7; 7; 8]
```

```
[ 2  3  5  7;  
 -2  1  0  7;  
  5  2 -1  8 ]
```

```
b= (m>3) .* m  
   [ 0  0  5  7;  
     0  0  0  7;  
     5  0  0  8 ]
```