

5.11 → add figure(1) before subplots
6.10 → add figure(2) before plot
OR 6.10 plot goes in last subplot of 5.11

6.20 given

commonbar = [4 6 10] ← & DO NOT HAVE TO BE INTEGERS UNLESS USED AS ARRAY INDICES, SUBSCRIPTS

for hvar = commonbar

~
~

for i = 1:10

for k = 1:2:11

for i = 1:10

a(i) = i^2

end.

LOOP VARIABLE USED AS ARRAY SUBSCRIPT

LOGICAL OPERATORS

if (a == 5 && b == 6) && || ← use Comparing scalars

if (array A == [1 2] & array B == [3 4]) & | ← " " arrays.

FIND COASTLINE

q. data in 2D array.

$i = \text{find}(\text{row} == 0)$

RETURNS INDICES OF 2D ELEMENTS
WHOSE VALUES == 0
- HERE AS "LINEAR" INDEX VALUES

→ HERE AS ROW & COL INDEX

$[\text{row}, \text{col}] = \text{find}(\text{row} == 0)$

$a = [1\ 0\ 2; 0\ 3\ 4] \rightarrow \begin{bmatrix} 1 & 0 & 2 \\ 0 & 3 & 4 \end{bmatrix}$

$\gg i = \text{find}(a == 0)$

$\gg i = 2$
 3

$\gg [\text{row}, \text{col}] = \text{find}(a == 0)$

row = 1
2

col = 2
1

q. find smallest value in array.

$\gg \text{min}(a)$

$\gg \text{ans} = 0\ 0\ 2$ ← returns min of each col

$\gg \text{min}(\text{min}(a))$

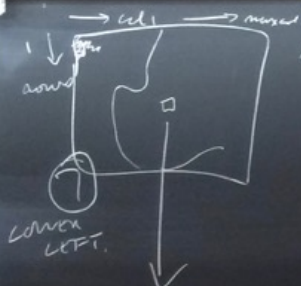
$\gg \text{ans} = 0$

o/d SIMILAR FOR MAX

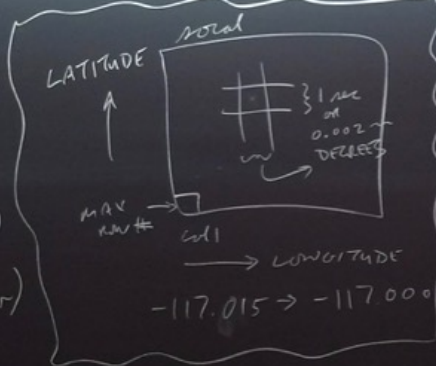
$\gg \text{lowElev} = \text{min}(\text{min}(\text{row}))$

$\gg \text{row}(i) = \text{lowElev}$

↳ where $i = \text{find}(\text{row} == 0)$ o/d CONST
 $\gg \text{row}(\text{row}, \text{col}) = \text{lowElev}$



image(social, 'CData Mapping', 'scaled')



$$hi_Elev = \max(\max(\text{social}))$$

$$i = \text{find}(\text{social} == hi_Elev)$$

$$\text{social}(i(1)) = \text{low_Elev}$$

$$\text{social}(\text{row}(i)-5 : \text{row}(i)+5, \text{col}(i)-5 : \text{col}(i)+5) = \text{low_Elev}$$

FROM $[\text{row}, \text{col}] = \text{find}(\text{social} == hi_Elev)$

GET LAT & LONG OF HI & LO PTS.

KNOW LAT & LONG OF LOWER LEFT CORNER OF SOCIAL

$$[r, c] = \text{size}(\text{social})$$

VALUE OF r IS # OF LAST ROW IN SOCIAL

∴ KNOW LAT & LONG @ SOCIAL(r, 1) % LOWER-LEFT
 $\approx 32.4 \quad -117.4$

ALSO IN HEADER cellsize = $\frac{1}{3600} = 1/60/60$
 ∴ " IS DECIMAL EQUIV DEGREES OF 1 SEC.

GAME

```
countGuessTotal = 0; ← INITIALIZE COUNTERS  
countGames = 0; ←
```

```
while play == true
```

```
  % play a game
```

```
  countThis = 0; ← INITIALIZE THIS COUNTER INSIDE GAME LOOP
```

```
  while guess... (is wrong)
```

```
    % get guess
```

```
    countThis = countThis + 1; ← INCREMENT COUNTER
```

```
  end
```

```
  % end this game
```

```
  countGames = countGames + 1; ←
```

```
  countGuessTotal = countGuessTotal + countThis ← ACCUMULATE SUM
```

```
end
```

```
% STOP PLAYING
```

```
aveGuessPerGame = ...
```