BA – Discussion #2, 2020-09-11

Nice to see you again!

- Devin Goodwin
  devinbg@bu.edu

- Leah Gaeta
  ltgaeta@bu.edu

*please unmute, show video, and fully participate

Announcements

- make sure you have all three exam and all quizzes listed in the syllabus on your calendar
  → first exam is 3 weeks from today!

- if you have questions, please check to see if your question has been answered on the discussion forum, and if it hasn't then post it! (Do not email Prof. Attaway)

- TA Open Hours:
  → Sunday 4-10 pm
  → Tuesday 6-10 pm
  → Thursday 4-10 pm (Devin's OH: 6-8 pm Thur)

- Practice quiz this afternoon ... please take like a real quiz!!!
  → it will release on Gradescope at 4:40 pm, due by 4:55 pm
  → quizzes will only be accepted via gradescope, not email
  → download quiz, fill it out, scan it, upload as a PDF.
    - .pdf must be correct number of pages
    - time yourself so that you're ready for week
  → will also be open for a couple days to get extra practice with uploading

- Classes next week:
  → in-person will be an option by invitation only
  → class will be conducted the same, so fully via zoom with breakout rooms like we have been doing
  → you can only come to the room if you received an invite

Review of Material

- vectors and matrices
- matrix operations and functions
- logical indexing vs. using find()
- anything else?

Vectors & Matrices

```
v = 1:5
v = 1  2  3  4  5
v(2:4) = 7:9
v = 1  7  8  9  5
```
*replaces middle 3 elements

length() & size()
→ use length for vectors
→ use size() for matrices
can use for vectors to see if row or column

colon operator
```
1:0.5:3
1  1.5  2  2.5  3
```

linspace(1, 3, 5)
↑ # of elements

spacing for linspace:
$(x_2 - x_1)/(n-1)$

ex. linspace(9, 5, 3)
$(5-9)/(3-1)$
$-4/2 \ldots -2$
```
>> 9  7  5
```

Other

rand() a random number in the range of 0 to 1
rand()*5 + 5 will generate a # from 5-10

formula for rand():
rand() * (MAX - MIN) + MIN

isequal() vs. ==

```
vec1 = 1:3;
vec2 = [1 0 3];
isequal(vec1, vec2)
   >> 0
vec1 == vec2
   >> 1  0  1
```

scalar multiplication

```
>> [5 3 0] * 5
   >> 25  15  0
```

logical indexing vs. using find()

```
>> vect = randi([-5,10], 1, 6)
vect =
   -5  8  9  5  7  6

>> vect(vect > 0)
vect =
   8  9  5  7  6
```
gives the actual values that are > 0

```
>> find(vect > 0)
ans =
   2  3  4  5  6
```
returns the indices of values > 0

Array multiplication

$A = \begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix}$ (2×2) $B = \begin{bmatrix} 2 & 1 & 3 \\ 1 & 5 & 6 \\ 3 & 6 & 0 \end{bmatrix}$ $C = \begin{bmatrix} 3 & 2 & 5 \\ 4 & 1 & 2 \end{bmatrix}$ (2×3)

ex.

$3 * A \Rightarrow \begin{bmatrix} 3 & 12 \\ 9 & 6 \end{bmatrix}$

[ ] = brackets, used to create arrays i.e. vector and matrices

( ) = parantheses, used for argument

$A * C$

note that C * A does NOT work!

$\begin{bmatrix} 1 & 4 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 3 & 2 & 5 \\ 4 & 1 & 2 \end{bmatrix}$

inner dim. (2×2 and 2×3; match, so valid!)

2 × 3

$\begin{bmatrix} 1*3+4*4 & 1*2+4*1 & 1*5+4*2 \\ 3*3+2*4 & 3*2+2*1 & 3*5+2*2 \end{bmatrix}$

$\begin{bmatrix} 19 & 6 & 13 \\ 17 & 8 & 19 \end{bmatrix}$ ... 2×3!

```
v = 1:5
delete the 3rd element
>> v(3) = [];
>> ans =
     1  2  4  5
```

Element by element multiplication

$D = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ $E = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$

```
>> D .* E
>> ans =
     0   6
    14  24
```