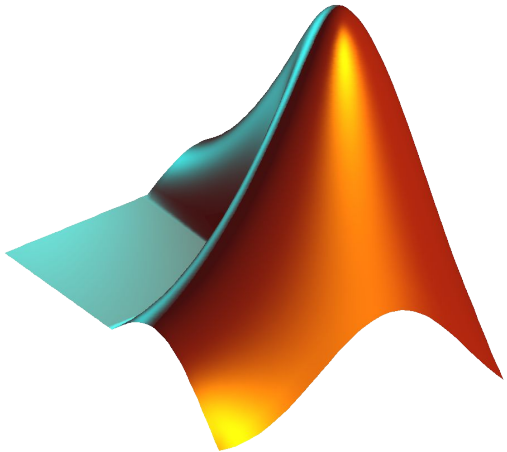
CS 1112 Introduction to Computing Using MATLAB

Instructor: Dominic Diaz



Website: https://www.cs.cornell.edu/courses/cs1112/2022fa/

Today: Vectors (1D arrays)

Agenda and announcements

- Last time
  - Finished up on functions
- Today
  - Vectors (1D arrays)
- Announcements
  - Project 3 released tomorrow
    - Partner matching survey open (submit only if you would like us to suggest a partner to you).
  - Project 1 grades released yesterday
    - Submit a regrade request if you think you were graded unfairly
  - Discussion worksheets will mostly be checked off using MATLAB Grader from now onwards (but there might be one or two more worksheets that your TA will need to check off).

Catching con artists

Say we're approached by a group of people asking if we want to roll a die for money. If we land on 1, 2, 3, or 4, they give us $5 but if we land on 5 or 6 then we give them $5.

Should we do it?

No, because the dice might be unfair!

For an unfair dice, the probability of landing on each side could be different! They could have rigged the dice so that it lands on 5 or 6 many more times than 1-4.

What can we do to make sure the die is fair?

Roll the dice a bunch of times!

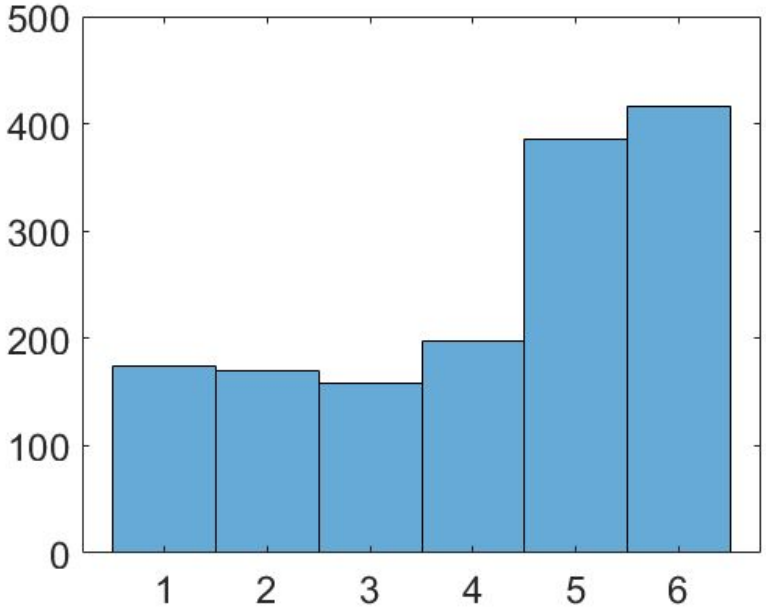
Rolling a die N times…

5	2	1	6	5	5	3	1	6	5

1st roll 2nd roll 3rd roll …

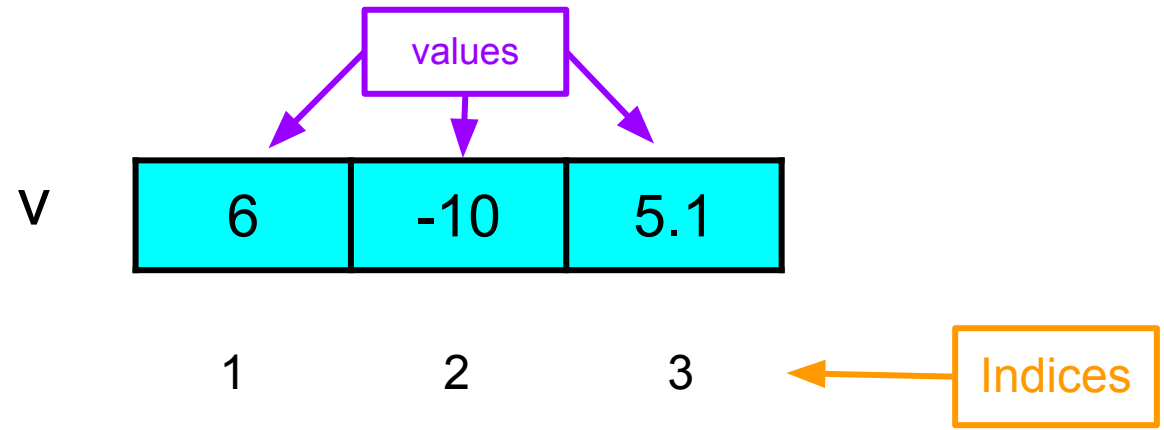
Using the results of these N rolls, we could plot the results in a histogram!

To plot a histogram, we need to store all rolls and we can do this using a vector!



1D Array: Vector

- An array is a collection of like data organized into rows and/or columns
- A 1D array is a single row or column, also called a vector
- An index identifies the position of a value in a vector



Many different ways to create a vector

```
v = zeros(1,6);
```

Similar functions: `ones`, `rand`

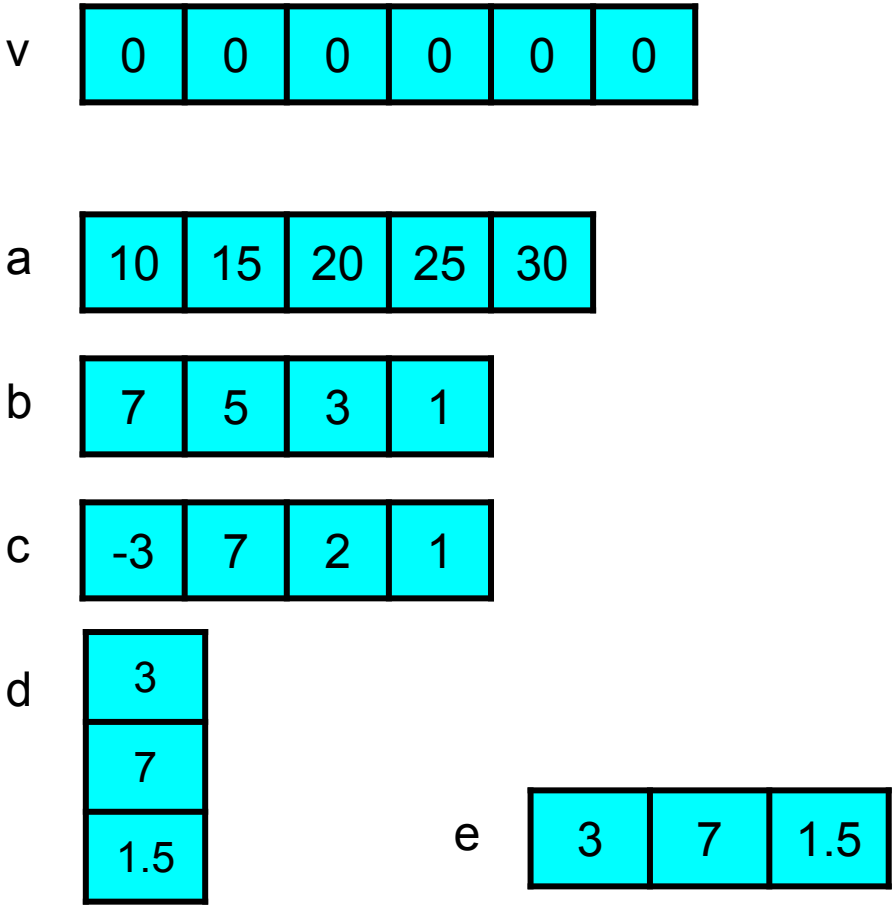
```
a = linspace(10, 30, 5);
```

```
b = 7:-2:0;
```

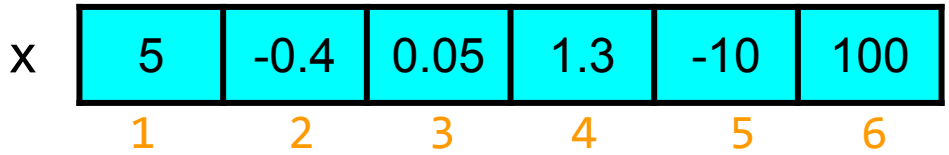
```
c = [-3 7 2 1];
```

```
d = [3; 7; 1.5];
```

```
e = d';
```



Array indexing starts at 1



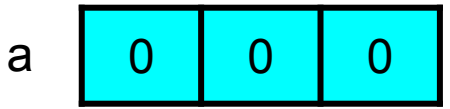
Let k be the index of vector x, then

- k must be a positive integer
- 1 <= k <= `length(x)`
- To access the kth element, use: `x(k)`

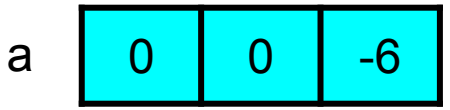
`disp(x(1))` would display the number 5, because we have asked the computer to display the 1st number in the vector

Changing values in a vector

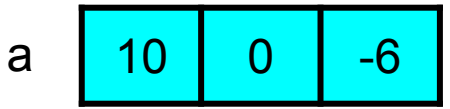
```
% Create a vector with 3 zeros
a = zeros(1, 3);
```



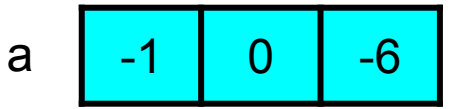

```
% Change the 3rd element in the vector to -6
a(3) = -6;
```




```
% Change the 1st element in the vector to 10
a(1) = 10;
```




```
% Change the 1st element in the vector to -1
a(1) = -1;
```



For loop pattern for processing a vector

MATLAB `length` function: input can be a vector and the output is how many elements are in the vector.

```
% Given a vector v

for i = 1:length(v)

    [Code doing something
     to or with v(i)]

end
```

```
% Example: sum all values in a vector v
s = 0;
for i = 1:length(v)

    s = s + v(i);

end
```

10	3	-2	1.5	-0.5	2

Example: sum all values within a vector

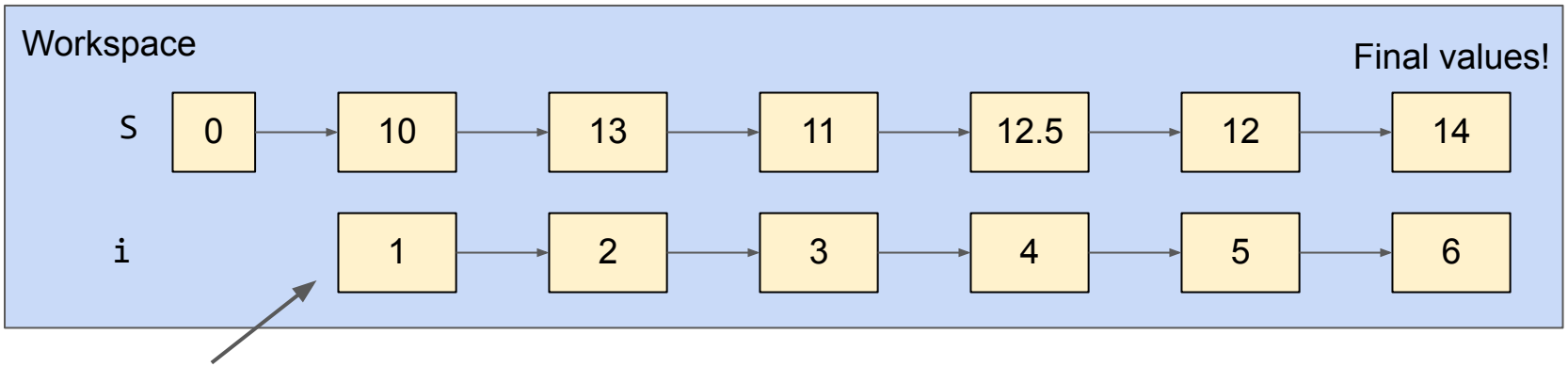
```
% Example: sum all values in a vector v
s = 0;
for i = 1:length(v)

    s = s + v(i);

end
```

10	3	-2	1.5	-0.5	2

sum = 10 + 3 - 2 + 1.5 - 0.5 + 2 = 14



Once MATLAB starts the `for` loop, `i` is created.

Another loop example with vectors

```
% count number of odd numbers in vector v

numOdd = 0;
for k = 1:length(v)
    if rem(v(k), 2) == 1
        numOdd = numOdd + 1;
    end
end
```

If `v(k) = 13`,
`remainder(13/2)` would equal `1`
So `rem(v(k), 2) == 1` is true and thus 1 would be added to `numOdd`

If `v(k) = 100`,
`remainder(100/2)` would equal `0`
So `rem(v(k), 2) == 1` is not true and nothing would happen for the kth iteration of the for loop.

…

13	100	2	1	16	-25

Be careful not to overwrite MATLAB functions

```
% Example: sum all values in v
sum = 0;
for i = 1:length(v)

    sum = sum + v(i);

end
```

Not good because this code overwrites the sum function

```
% Example: sum all values in v
s = 0;
for i = 1:length(v)

    s = s + v(i);

end
```

There are a few functions that have been covered in the class that you want to make sure you don't overwrite when defining variables:

max, min, sum, length, size, rem, zeros, ones, double, rand, abs, floor, ceil

Do not overwrite on exams

Do not overwrite on projects

Back to rolling a die N times…

```
% Roll a die N times and plot results in a histogram
N = 1000;
dieRolls = zeros(1,N);

% roll a die N times and record the result




% plot histogram of dice rolls
```

Back to rolling a die N times…

```
% Roll a die N times and plot results in a histogram
N = 1000;
dieRolls = zeros(1,N);

% roll a die N times and record the result
for i = 1:N
    currentRoll = ceil(rand*6);

end

% plot histogram of dice rolls
histogram(dieRolls)
```

Back to rolling a die N times…

```
% Roll a die N times and plot results in a histogram
N = 1000;
dieRolls = zeros(1,N);

% roll a die N times and record the result
for i = 1:N
    currentRoll = ceil(rand*6);
    dieRolls(i) = currentRoll;
end

% plot histogram of dice rolls
histogram(dieRolls)
```

Change the `i`th element of `dieRolls` to the value stored in `currentRoll`.