While you are waiting

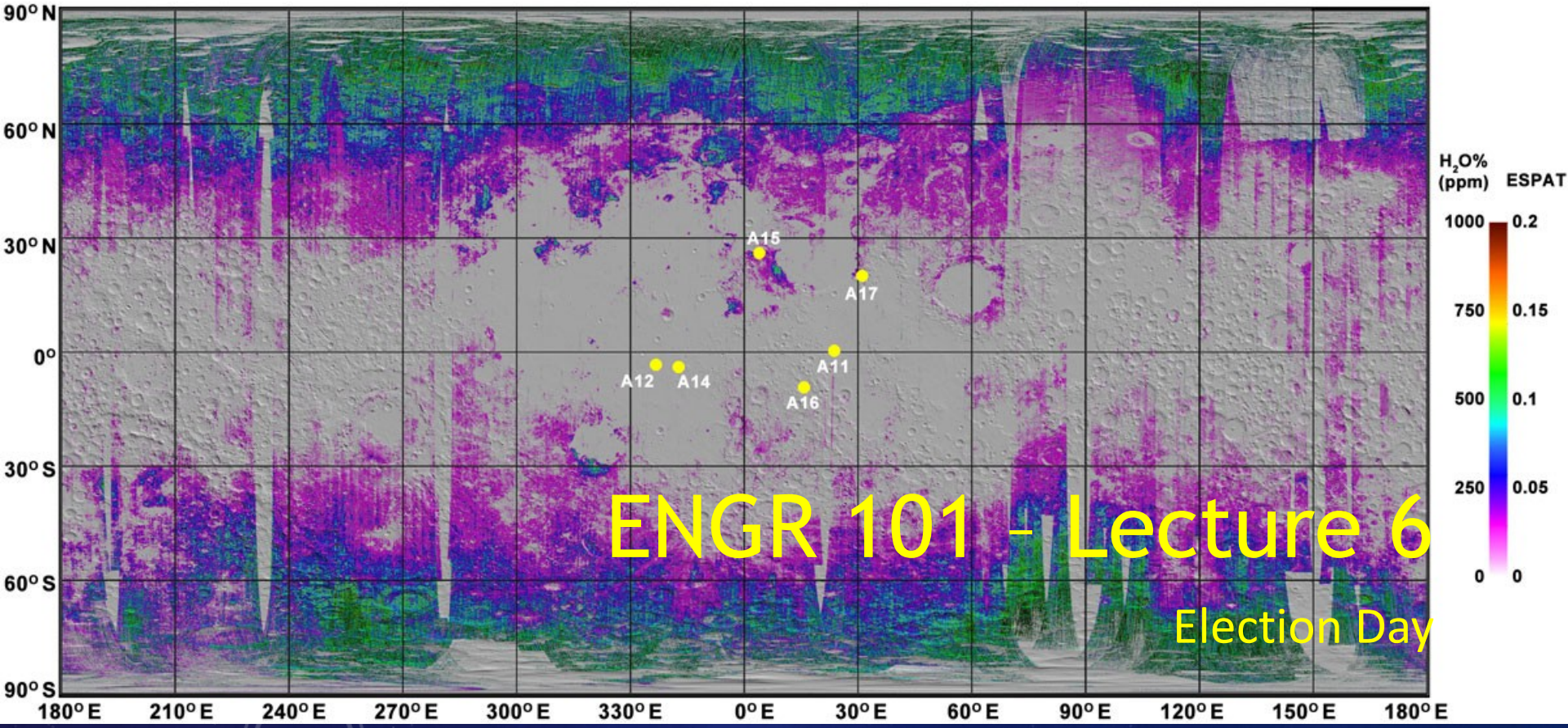
MICHIGAN ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

Did you know…

If you do a Google search for

"**do a barrel roll**"

the entire page will rotate



ENGR 101 – Lecture 6

Election Day

Laura Alford, James Juett, Rick Niciejewski 9/25/17

Today's Lesson Plan

- Use built-in MATLAB functions to select elements from arrays
- For help with a MATLAB function, use help in the command window

For example,

>> help max

- To answer exercises, work as a group using MATLAB on your laptop !!!
  - Use the command window for the exercises

Today's Lesson Plan

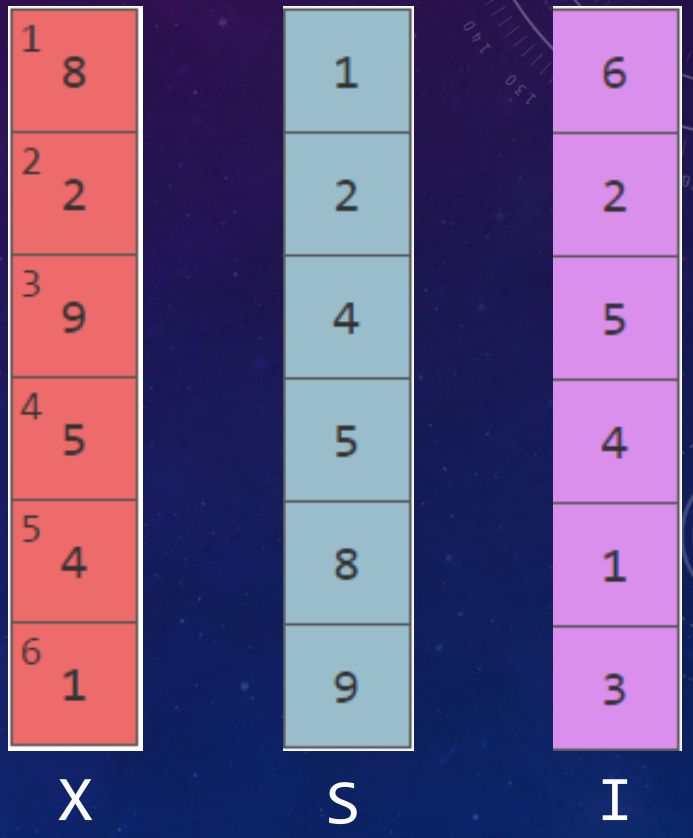
- Also, you may notice that the solutions are included with the .pdf in 00_Todays_lecture

- Our goal is that you understand how to reach these solutions. You will have time to input them into MATLAB and discuss amongst yourselves or with one of us during the exercise breaks

The sort Function

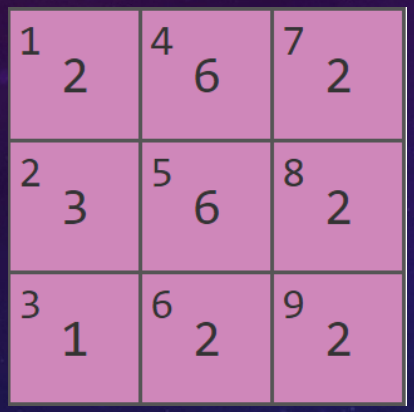
- By default, the sort function works with column vectors. (If you have a matrix, each column is sorted individually.)
- sort uses a compound return to give us both a sorted version of the vector AND a vector of sorted indices.
- You can provide 'ascend' or 'descend' as a another argument to specify order.

```
[S, I] = sort(X, 'ascend')
```

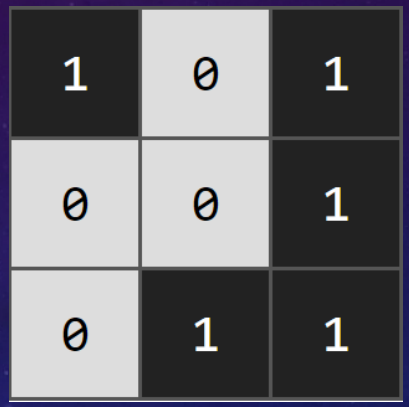


The `find` Function

- The `find` function converts a logical index matrix into a regular index matrix.



`x`

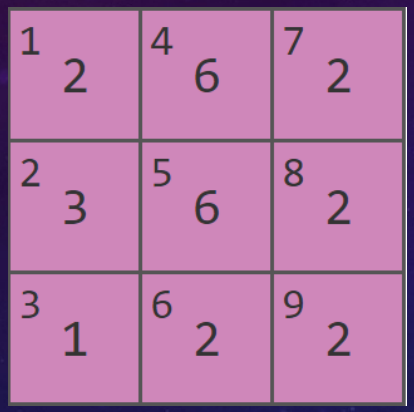


`x == 2`

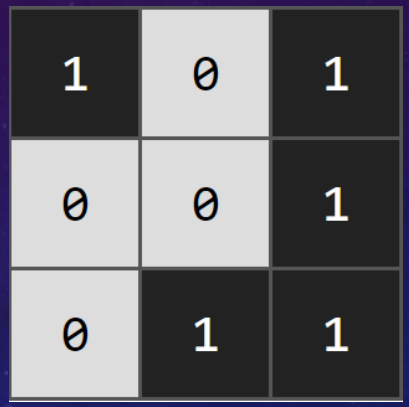
`find(x==2)`

The `find` Function

- The `find` function converts a logical index matrix into a regular index matrix.



`x`



`x == 2`

`find(x==2)`

- You can do conditional selection with either.
  - e.g. `x(x==2)` and `x(find(x==2))` are equivalent.

Plain logical indexing is simpler and usually more efficient.

More MATLAB Functions, `all`

- The `all` function returns `true` if ALL elements of an array are `true`. If even one element is `false`, it will return `false`.

EXAMPLE

>> A = [6 2 15 9 7 11]; B = [6 2 15 9 0 11];

>> `all`(A) will give ans = 1

>> `all`(B) will give ans = 0

- Use `all(all(A))` if you are working on a matrix

More MATLAB Functions, any

- The any function returns `true` if ANY element of an array is `true`. It only returns `false` when all elements are `false`.

EXAMPLE

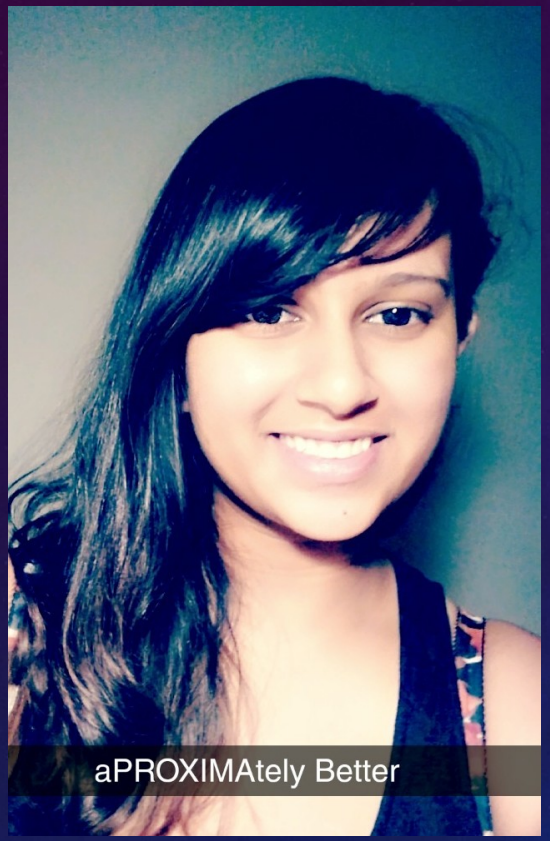
>> A = [6 0 15 0 0 11]; B = [0 0 0 0 0 0];

>> any(A) will give ans = 1

>> any(B) will give ans = 0

- Use any(any(A)) if you are working on a matrix

Electing the Future President of Proxima b



Ishika Paul

Amit Shah

Saving and Loading a Workspace

- You can save all the variables in your MATLAB workspace to a file using the `save` command.

```
save('election');
```

We did this earlier to prepare a workspace for you.

- To restore a saved workspace, use the load command.

```
load('election');
```

You should do this now to load the workspace for this lecture.

- You can find the file 'election.mat' on 00_Todays_Lecture
  - Move it from your temp area to your MATLAB folder

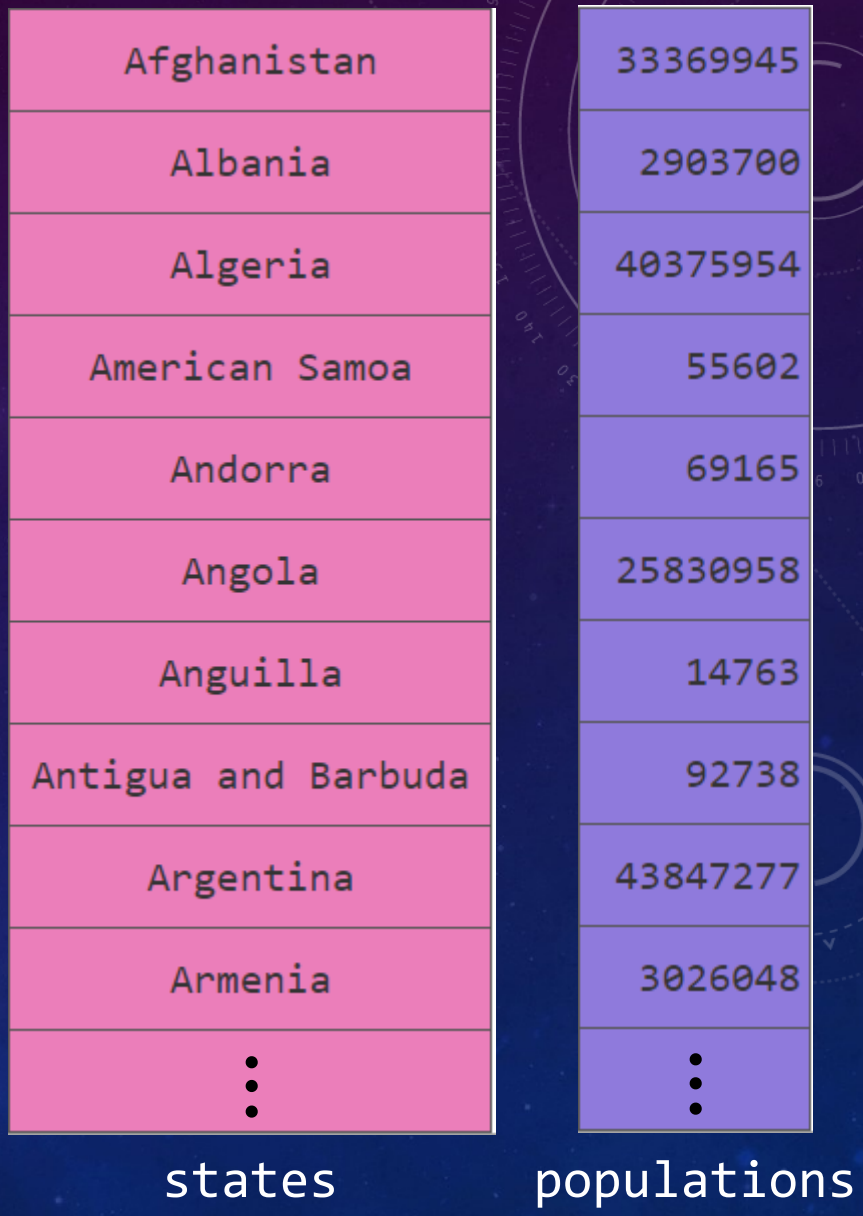
Number Formatting

- Before continuing, run the following command to ensure MATLAB prints the population numbers properly:

```
format long g
```

Population Data

- First, let's focus on two variables from the election workspace we just loaded:
- `states`
  This is a column vector containing a list of the states of Earth in ascending alphabetic order. This variable actually contains a different "type" of data called a `string` that is used to represent words.
- `populations`
  A parallel column vector to states that stores the population of the corresponding state.



5 min

Your turn: Population Data

- Write MATLAB expressions involving the `states` and `populations` vectors to compute answers to the following questions about the population data…

  - How many states are there with population less than 10 million?
  - What is the population of the largest state? …and which state is it?
  - Are there any states with less than 2000 people?
  - Which states have a population of more than 100,000,000?
    - *(Hint: Use a comparison on `populations` to generate a logical matrix, but then use it to index into `country` instead.)*

Solution: Population Data (*use PowerPoint only)*

- How many states are there with population less than 10 million?

```
sum(populations < 10000000)
length(find(populations < 10000000))
```

- What is the population of the largest state? Which is it?

```
[p, i] = max(populations);
display(states(i));
```

- Are there any states with less than 2000 people?

```
any(populations < 2000)
```

- Which states have a population of more than 100,000,000?

```
states(populations > 100000000)
```

How many votes should each state get?

- Proposal: Assign the votes proportionally.
  Concern: Do the largest few states have most of the votes?
  - Find the total population of the 10 largest states. Compare this with the population of the rest of the World.
  - Also obtain a list of the 10 most populous states.

- *Hint: Use `sort` to get both the sorted populations, but also `I`, which you can use to sort states as well.*

```
[sortedPopulations, I] = sort(populations, 'descend');
```

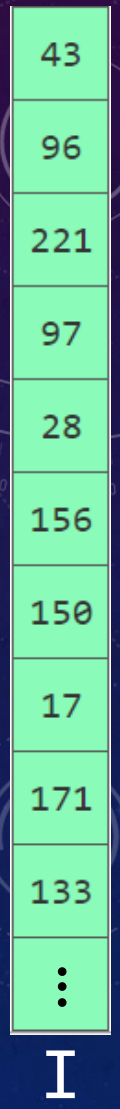
- See next slide for an example…

Sorting Parallel Vectors

- We already have sorted indices from the `populations` vector…
- Since the `states` vector is parallel, we can sort it just by putting it in order according to these indices.

```
[sortedPopulations, I] = sort(populations, 'descend');
sortedStates = states(I);
```

- From now on, when we want states ordered by population, we can just use the sorted vectors.
- However, we still have access to the original (the `sort` function produces a sorted copy – it does not change the original).

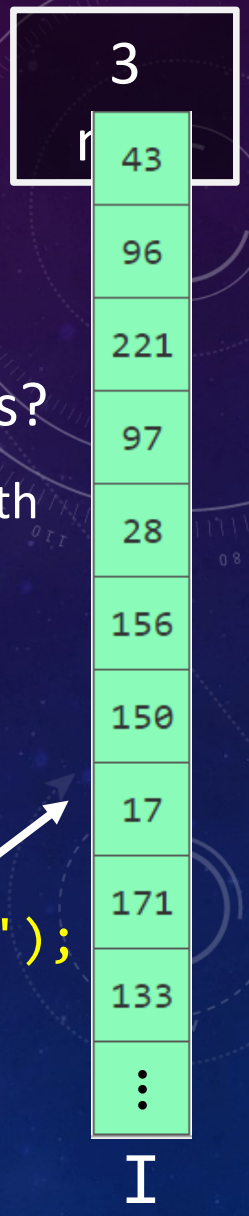


Your turn: How many votes should each state get? (*wait*)

- Proposal: Assign the votes proportionally.
  Concern: Do the largest few states have most of the votes?
  - Find the total population of the 10 largest states. Compare this with the population of the rest of the World.

- *Hint: Use this statement with sort to get the indices of our states sorted from most to least populous.*

```
[sortedPopulations,I] = sort(populations, 'descend');
```

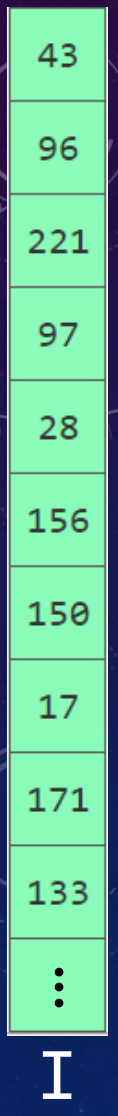


Solution: How many votes should each state get?

- Proposal: Assign the votes proportionally.
  Concern: Do the largest few states have almost all the votes?
  - Find the total population of the 10 largest states. Compare this with the population of the rest of the World.

```
[sortedPopulations, I] = sort(populations, 'descend');
sumTopTen = sum(populations(I(1:10)));
sumRest = sum(populations(I(11:end)));
TopTenStates = sortedStates(1:10);
```

sumTopTen	sumRest
4,318,189,480	3,114,473,795



3 min

Exercise: An Alternate Voting Scheme

- In order to ensure small states have some representation, an alternate voting scheme has been proposed:
  - All states get 10 votes.
  - For each full 10 million people, a state receives 1 additional vote.

- Under this scheme, compute the number of votes for each state, and store them in a column vector called `numVotes`.
  - `numVotes` should be "parallel" to the other vectors. That is, state names, populations, and number of votes correspond across rows.

- Hint: Use the `floor()` function to round down.

Solution: An Alternate Voting Scheme

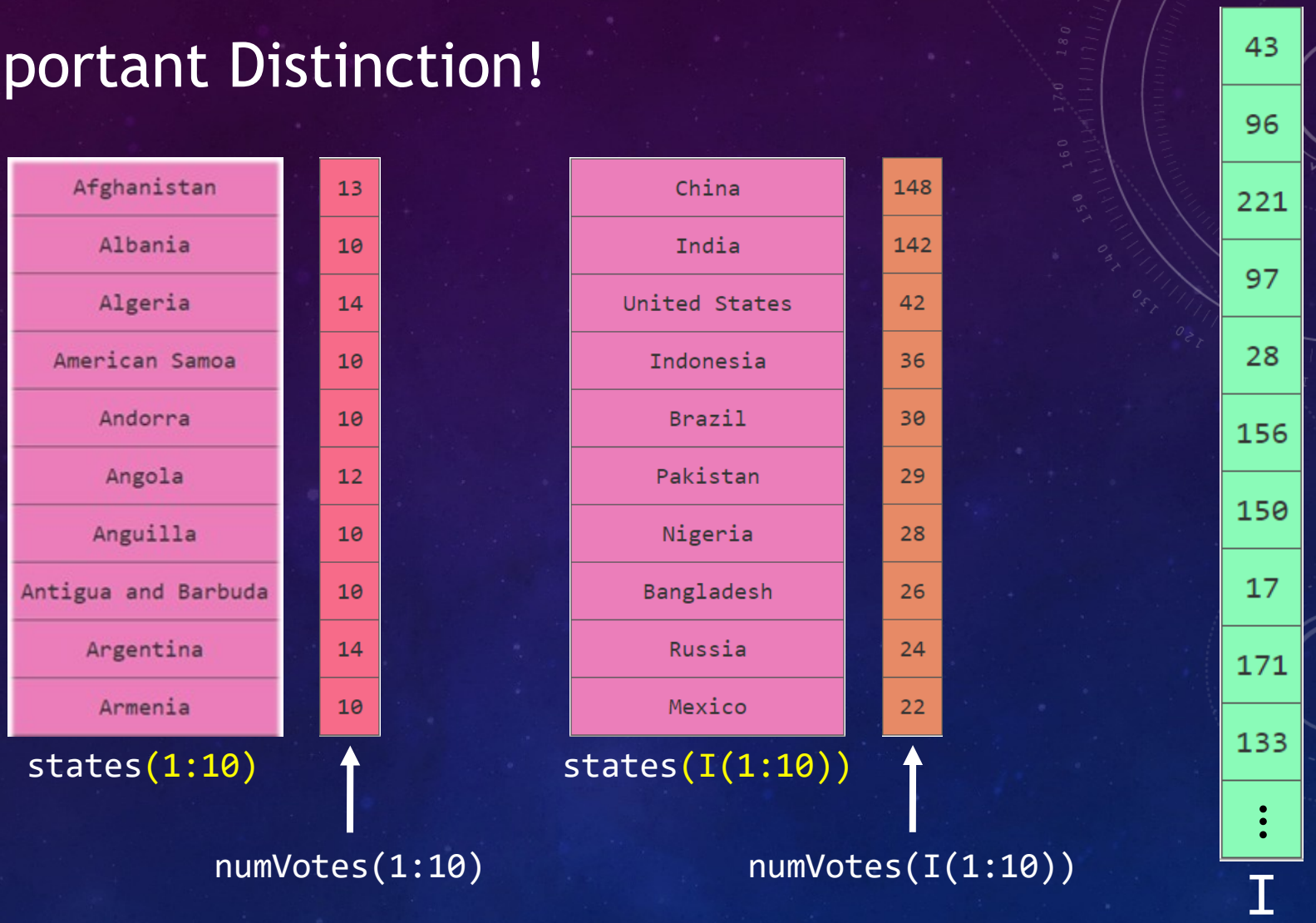
- All states get 10 votes.
- For each full 10 million people, a state receives 1 additional vote.

```
numVotes = 10 + floor(populations ./ 10000000)
```

states	populations	numVotes
Afghanistan	33369945	13
Albania	2903700	10
Algeria	40375954	14
American Samoa	55602	10
Andorra	69165	10
Angola	25830958	12
Anguilla	14763	10
Antigua and Barbuda	92738	10
Argentina	43847277	14
Armenia	3026048	10
⋮	⋮	⋮

sum(numVotes(I(1:10)))	527
sum(numVotes(I(11:end)))	2471
total	2998

Important Distinction!



Sorting Parallel Vectors

- We have sorted indices from the `populations` vector…
- Since the `states` vector is parallel, we can sort it just by putting it in order according to these indices.

```
[sortedPopulations, I] = sort(populations, 'descend');
sortedStates = states(I);
sortedNumVotes = numVotes(I);
```

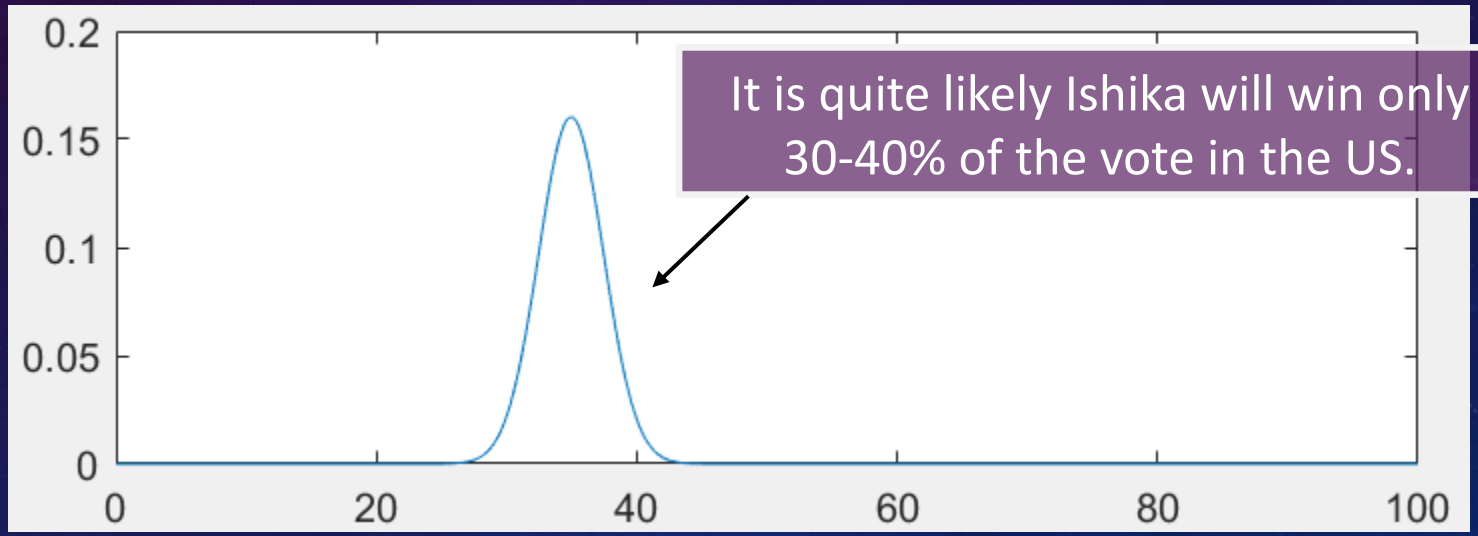
- From now on we can use the sorted vectors and we won't have to use `I` as an intermediary.
- We used new vectors here to create sorted copies but also to preserve the original ordering.

Break Time

We'll start again in 5 minutes.

Forecasting the Election

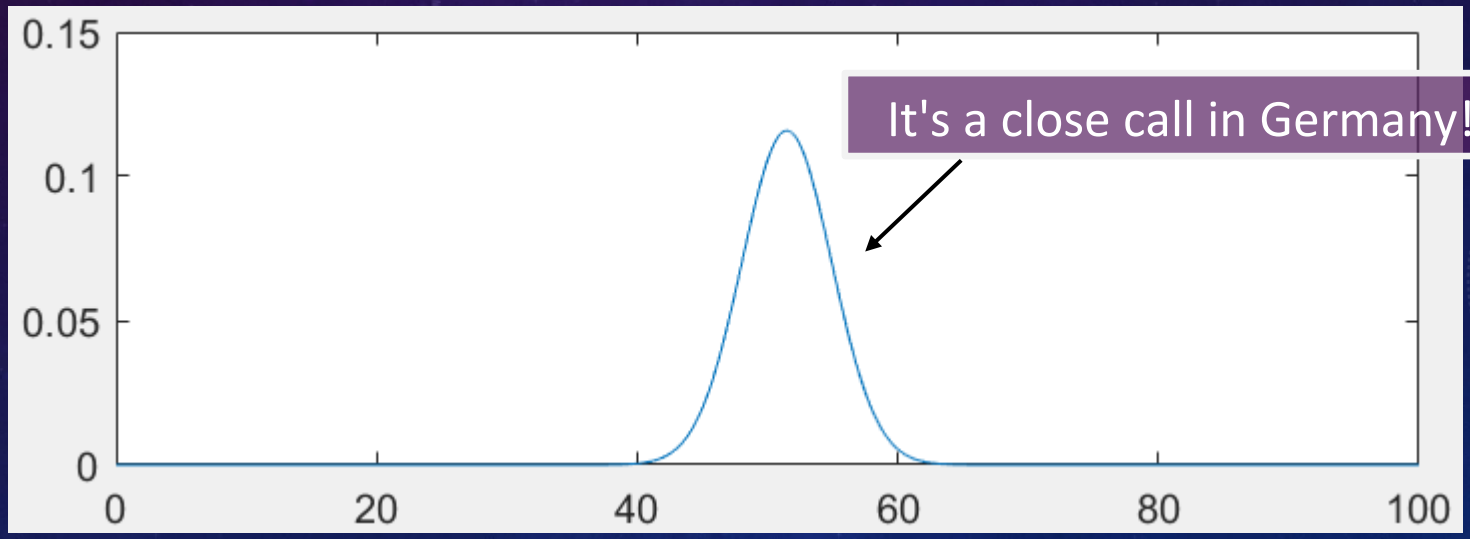
- Polling data from each state has been collected to gauge how likely candidates are to win those votes.
- From the data, we can model the probability of outcomes for each state as a normal distribution.



Polling distribution for Ishika in the US (mean = 34.06, stddev = 2.49)

Forecasting the Election

- We can model each country individually, but together there are over 200 different states each with their own distribution!
- Each also has a different number of votes! It's a nightmare to model the joint distribution of all of them mathematically.



Polling distribution for Ishika in Germany (mean = 51.48, stddev = 3.44)

A Computational Approach

- Instead of trying to model the distribution mathematically, we'll run many different simulations of the election.
- In a simulation, we "roll the dice" for each state individually, determine the winner for each, then tally up the votes.
  - We'll assume that the winner in a particular state will receive ALL of its votes ("winner take all").
- Then, we repeat the entire process several times.
  - Each simulation is a hypothetical run of the election.
  - Over time, we can see who is winning more simulations.
  - The percent of simulations won is our forecast for a candidate.

Simulating the Election in One State

- We can call the `randn()` function to sample from a standard normal distribution (mean = 0 and stddev = 1).
- Each call produces a different result!

randn()	randn()	randn()	randn()	randn()
-1.2368	0.2147	2.0108	0.0256	0.3083

- We can manually adjust the results for a custom distribution.

```
mean + stddev .* randn()
```

- For Germany, this would be

```
51.48 + 3.44 .* randn()
```

48.2524	57.2393	51.9100	53.3035	48.2049

Vectorization!

- The `randn()` function plays nicely with vectorized code.
- We can provide a size argument, and `randn()` will produce a matrix of that size containing random samples.

54.4179
52.8187
47.5034
51.6167
49.9299

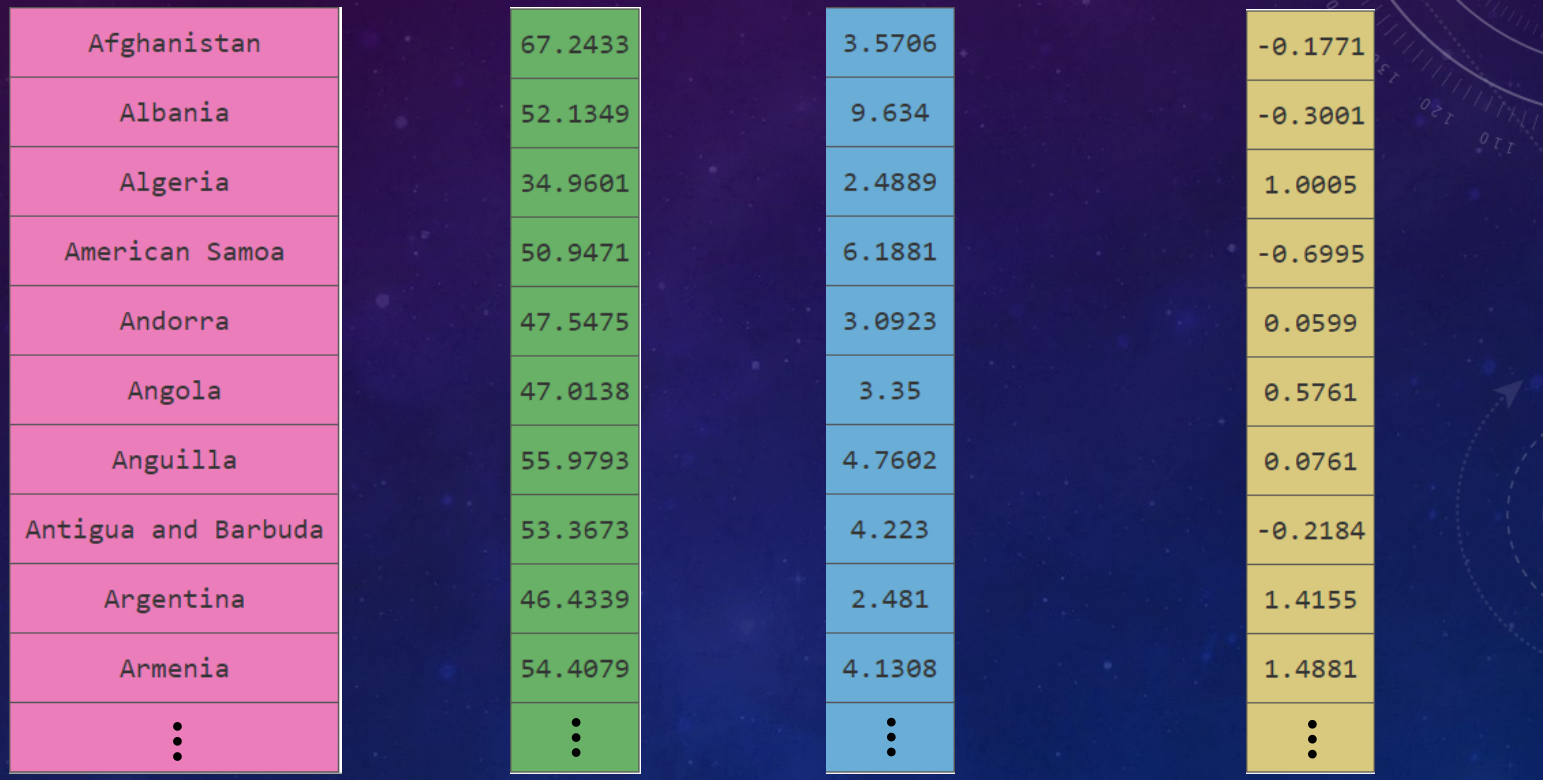
```
51.48 + 3.44 .* randn([5,1])
```

51.8558	47.9268	57.4817
50.6181	50.3679	47.4878
50.8267	54.1169	59.6583

```
51.48 + 3.44 .* randn([3,3])
```

Polling Data for Every State

- The `means` and `stddevs` variables provided with the election workspace are column vectors containing the parameters of the distributions for each state.



6 min

Exercise: Simulating the Election in ALL States

- You now have all the tools you need to simulate the election across all different states.
- Do this, and add up the simulated votes for Ishika.

1. Use the `means` and `stddevs` column vectors in the provided workspace, combined with a call to `randn( )` to simulate voting in all countries simultaneously. (tip: *vectorization slide)*
2. Use logical indexing to determine in which states Ishika received a majority of the votes. (tip: *win if number > 50)*
3. Sum up the votes for each state she won. (tip: *numVotes from alternate voting scheme)*
4. There are 2,998 votes total. Did she win?

Solution: Simulating the Election in ALL States

1. Use the `means` and `stddevs` column vectors in the provided workspace, combined with a call to `randn()` to simulate voting in all countries simultaneously.

```
outcomes = means + stddevs .* randn(size(states));
```

2. Use logical indexing to determine in which states Ishika received a majority of the votes.

```
wins = outcomes > 50; % this is a logical array
```

3. Sum up the votes for each state she won.

```
ishikaVotes = sum(wins .* numVotes);
```

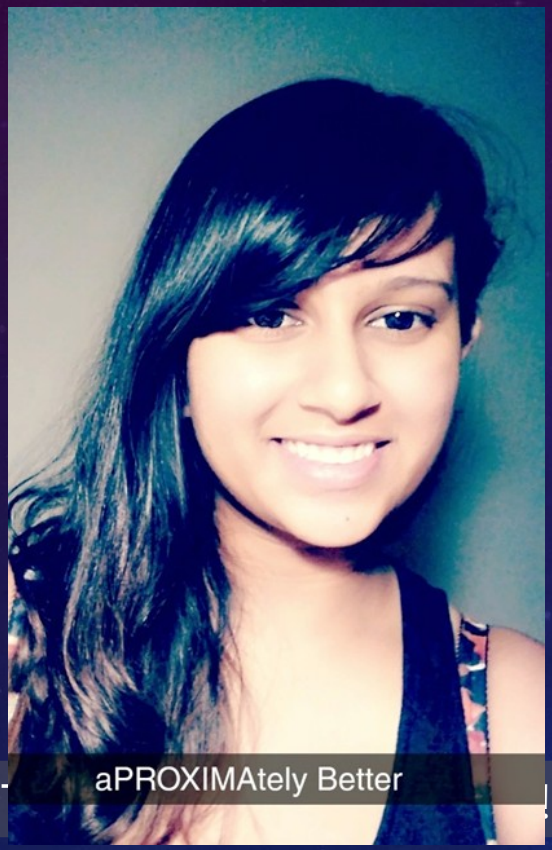
4. There are 2,998 votes total. Did she win?

```
ishikaVotes > sum(numVotes) ./ 2
```

Who Will Win?

- We ran 100,000 simulations….

55.87%



Ishika Paul

44.13%

Amit Shah

Challenge:

- It looks like Amit is just a bit less likely to win. In order to improve his chances, he wants to refocus his campaign.
- He plans to select the 5 closest (mean nearest to 50%) countries that have at least 15 votes and launch an advertising campaign targeting those peoples.
- If Amit can improve his mean polling percentage by 5 points in each of those countries, does this tip the overall election in his favor?
  - If not, what improvement would be necessary to do so?