YData: Introduction to Data Science

Class 17: Writing functions

Overview

Review of for loops

Conditional statements

Writing functions

If there is time: Introduction to statistical inference

- Parameters and statistics
- Sampling distributions

Reminder: keep working on your class project

A polished draft of the project is due on November 16^{th}

Also, homework 7 is due on Sunday November 2^{nd}

- I recommend finishing it early and then starting on your project by coming up with a topic and getting the relevant data.

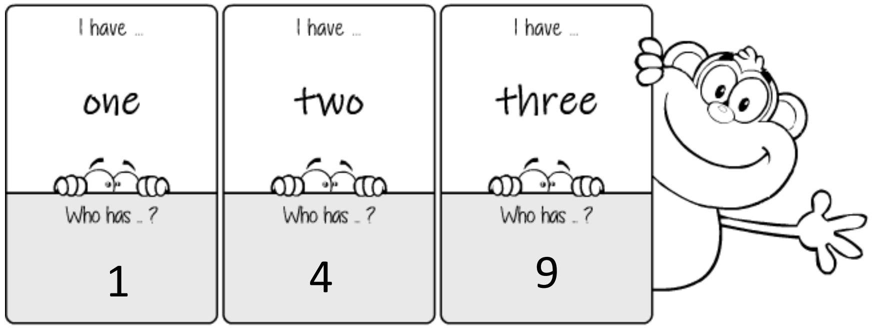
Quick review of for loops

Review: for loops

For loops repeat a process many times, iterating over a sequence of items

- Often we are iterating over an array of sequential numbers

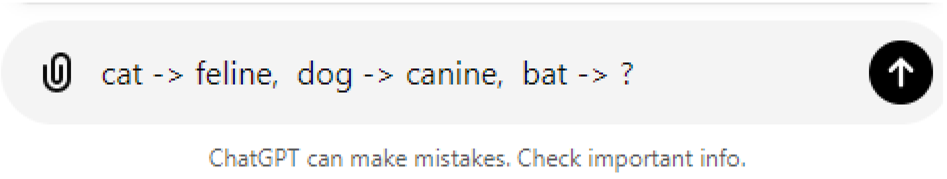
```
animals =  ["cat", "dog", "bat”]
for creature in animals:
    print(creature)

for i in range(10):
    print(i**2)
```



Review: enumerate and zip

We can use the enumerate() function to both items in a list, and sequential integers:

```
animals =  ["cat", "dog", "bat”]
        for i, creature in enumerate(animals):
                print(i, creature)
```



We can use the zip() function to get items for two lists:

```
        animal_order =  ["feline", "canine", "chiropteran"]
        for curr_order, curr_animal in zip(animal_order, animals):
                print(curr_order, curr_animal)
```

Let’s go over the temperature range example in Jupyter!

Conditional statements

Review: comparisons

We can use mathematical operators to compare numbers and strings

- Results return Boolean values True and False

Comparison	Operator	True example	False Example
Less than	<	2 < 3	2 < 2
Greater than	>	3 > 2	3 > 3
Less than or equal	<=	2 <= 2	3 <= 2
Greater or equal	>=	3 >= 3	2 >= 3
Equal	==	3 == 3	3 == 2
Not equal	!=	3 != 2	2 != 2

We can also make comparisons across elements in an array

Conditional statements

Conditional statements control the sequence of computations that are performed in a program

We use the keyword if to begin a conditional statement to only execute lines of code if a particular condition is met

We can use elif to test additional conditions

We can use an else statement to run code if none of the if or elif conditions have been met

```
num = 5
if num == 1:
    print("Monday")
elif num == 2:
    print("Tuesday")
elif num == 3:
    print("Wednesday")
elif num == 4:
    print("Thursday")
elif num == 5:
    print("Friday")
elif num == 6:
    print("Saturday")
elif num == 7:
    print("Sunday")
else:
    print("Invalid input")
```

Let's explore this in Jupyter!

Defining functions

Writing functions

We have already used many functions that are built into Python or are imported from different modules/packages

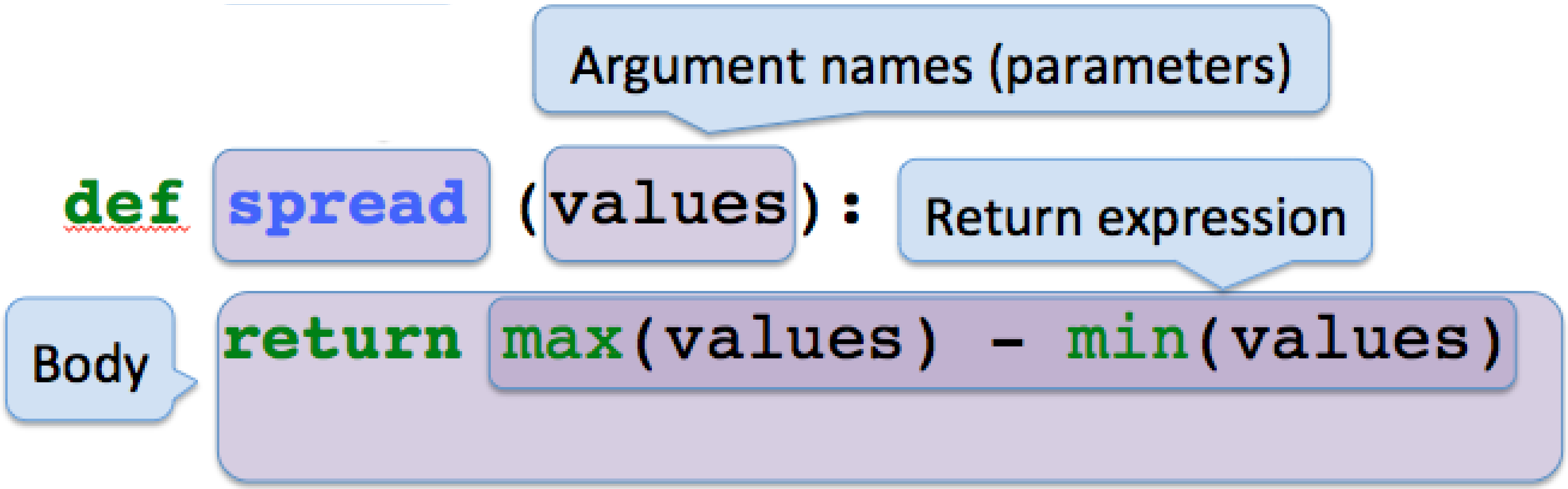
Examples…???

- sum()
- statistics.mean()
- np.diff()
- etc.

Let’s now write our own functions!

Def statements

User-defined functions give names to blocks of code



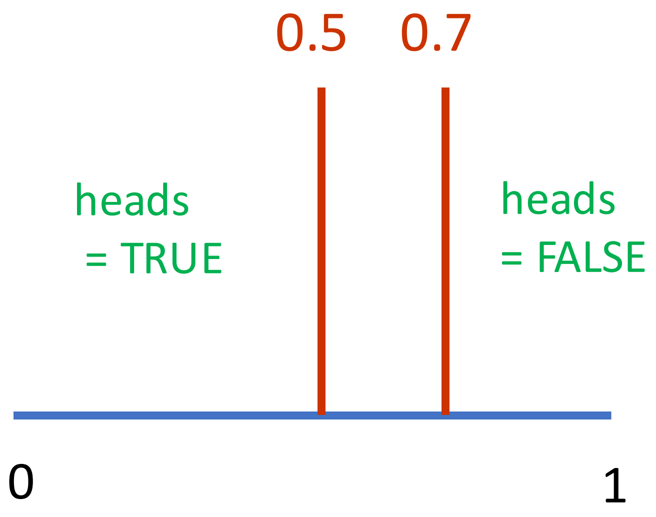
Let's explore this in Jupyter!

Practice: simulating flipping coins

Simulating flipping a coin

Let's practice writing functions by writing a function that can simulate flipping coins, where each coin has **π** probability of being heads

- Where **π** is a number between 0 and 1; e.g., **π** = 0.5 is a fair coin



We can do this using the following procedure:

1. Generate a random number between 0 and 1

```
rand_num = np.random.rand(1)
```

2. Simulate a fair coin (.5) by mark values less than .5 as heads (True)

```
heads = rand_num <=  .5
```

3. We can simulate a biased coin that will come up with heads 70% of the time (**π** = 0.7) using:

```
rand_num = np.random.rand(1)
heads = rand_num <=  .7
```

Simulating n random coin flips

We can simulate the number of heads we would get flipping a coin ***n*** times using:

1. Generate ***n*** random numbers uniformly distributed between 0 and 1

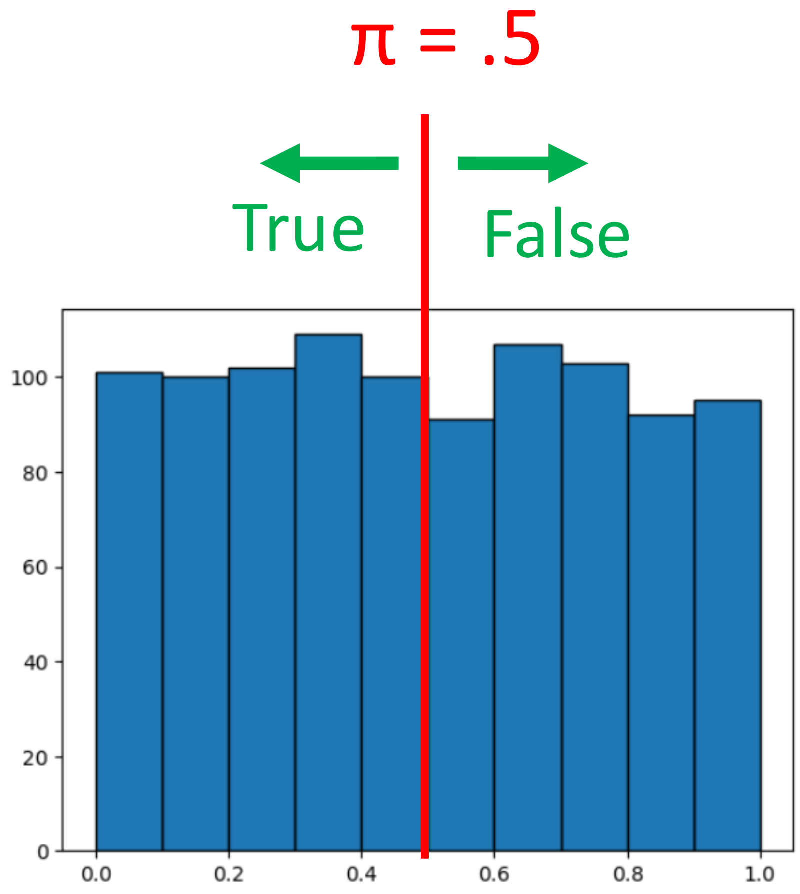
    `rand_nums = np.random.rand(n)`

2. Mark points less than π as being True, and greater π than as being False

    `rand_binary = rand_nums <=  prob_value`

3. Sum the number of heads (True's) we get

    `num_heads = np.sum(rand_binary)`



Let's explore this in Jupyter!

Statistical Inference

Statistical Inference

In **statistical inference** we use a sample of data to make claims about a larger population (or process)

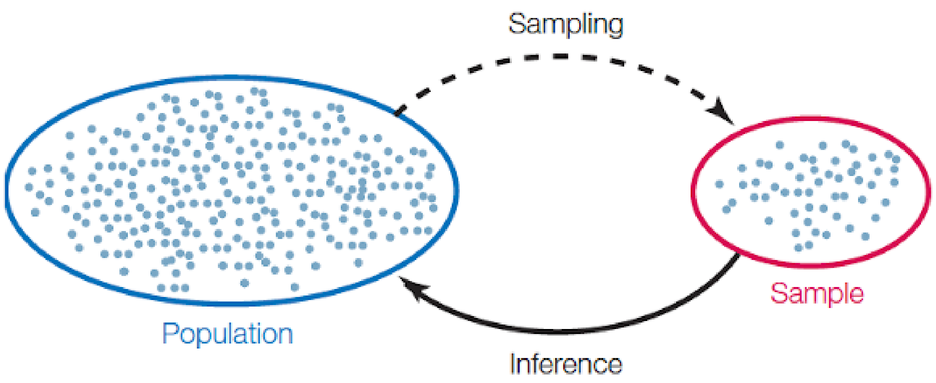
Examples:

<u>Voting data</u>

- **Population**: everyone who will vote
- **Sample**: survey of 1,000 randomly selected voters

<u>Bechdel data</u>:

- **Population**: All movies with budgets > $10,000,000
- **Sample**: 1794 movies randomly selected



Population: all individuals/objects of interest

Sample: A subset of the population

Terminology

A parameter is number associated with the <u>population</u>

- e.g., population proportion π
- e.g., the proportion of voters who will vote for Trump

A **statistic** is number calculated from the <u>sample</u>

- e.g., sample proportion $\hat{p}$
- e.g., proportion of 1,000 people in our sample

A statistic can be used as an estimate of a parameter

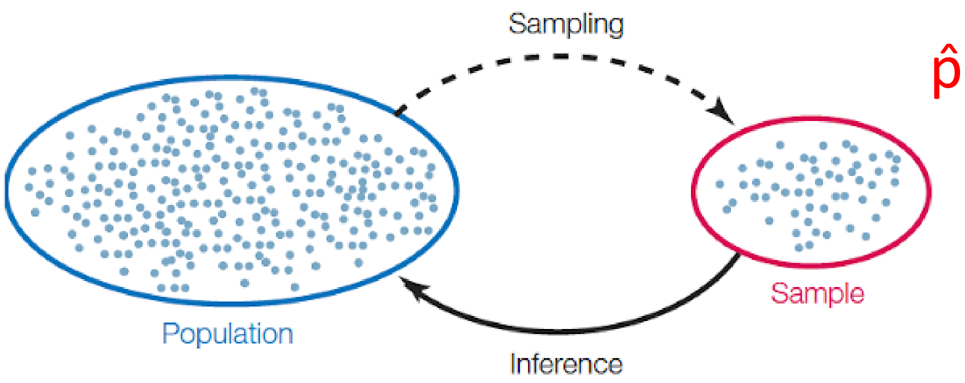
- A parameter is a single fixed value
- Statistics tend to vary from sample to sample

parameter

π

statistic

$\hat{p}$



Example:

- Using the proportion of 1,000 voters ($\hat{p}_{Trump}$) to estimate the proportion of all voters who will vote for Trump (π_{Trump})

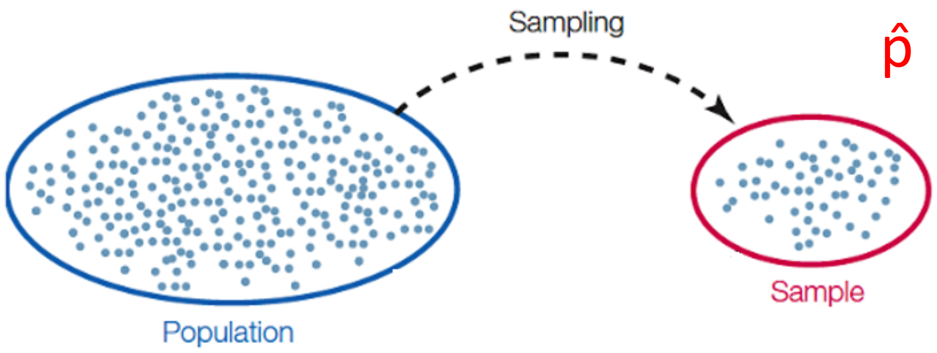
Examples of parameters and statistics

	Population parameter	Sample statistic	Bechdel example
Mean	μ	$\bar{x}$	statistics.mean(domgross_2013) $\bar{x}$ = \$95,174,783
Proportion	π	$\hat{p}$	bechdel.count("PASS")/len(bechdel) $\hat{p}$ = 0.45
Correlation	ρ	r	statistics.correlation(budget_2013, domgross_2013) r = 0.46

Sampling

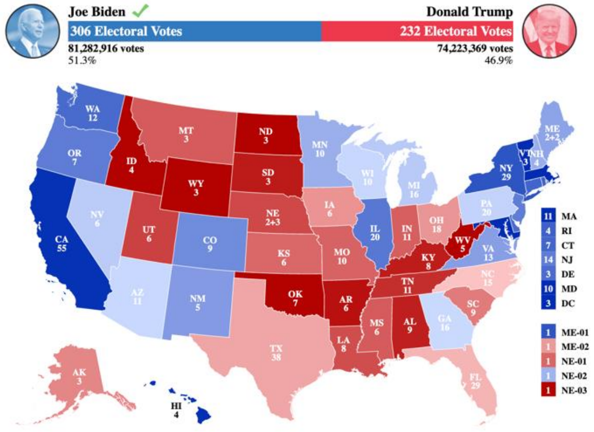
parameter π

statistic $\hat{p}$



Simple random sample: each member in the population is equally likely to be in the sample

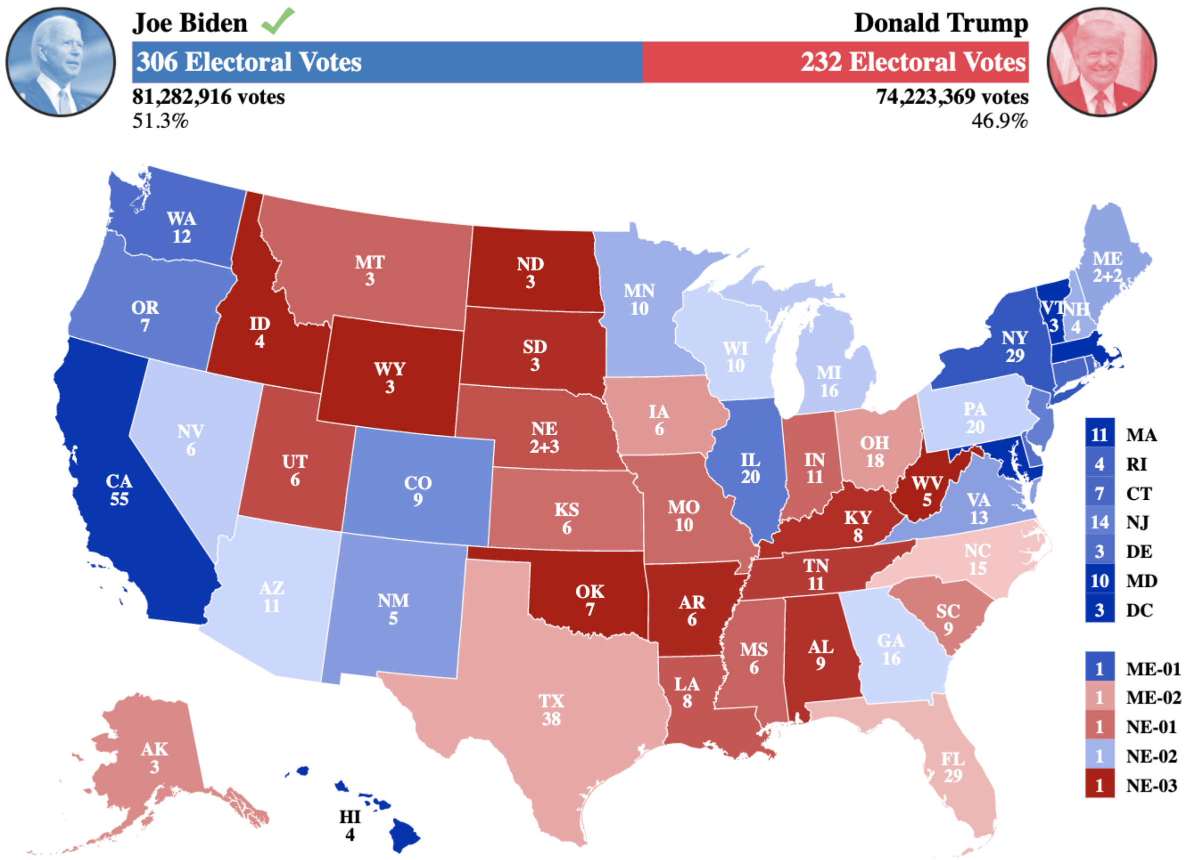
- Allows for generalizations to the population



Polls of 1,000 voters: $\hat{p}_{\text{Trump}}$

Vote on election day: π_{Trump}

Example: The 2020 US Presidential Election



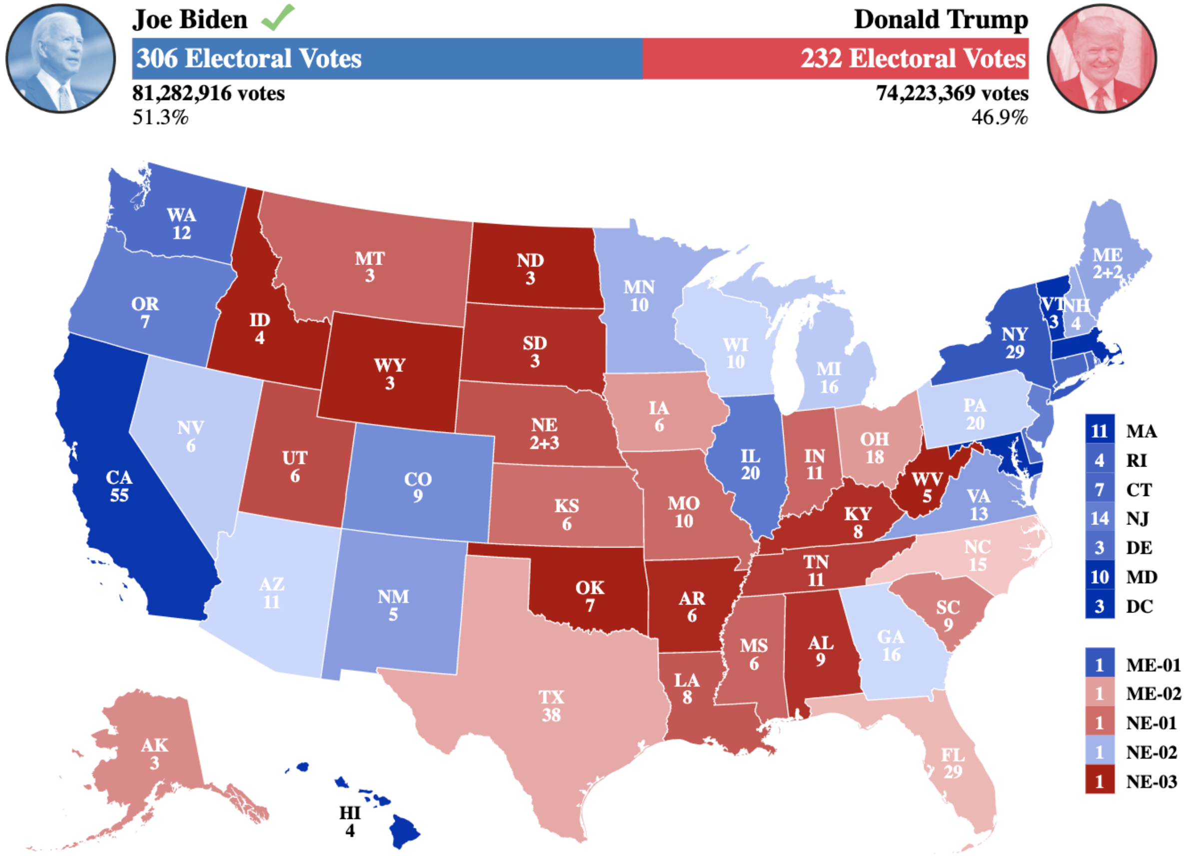
According to The Cook Political Report, the voting outcome in Georgia in 2020 was

- Trump = 2,461,854
- Biden = 2,473,633

We can denote the proportion of the vote that Trump got using π_{Trump}

- Q: what is the value of π_{Biden} ?

Example: The 2020 US Presidential Election



If 1,000 voters were randomly sampled, we could denote the proportion in the sample that voted for Biden using: $\hat{p}_{\text{Biden}}$

Would we expect $\hat{p}_{\text{Biden}}$ to be equal to π_{Biden}?

If we repeated the process of sampling another 1,000 random voters, would we expect to get the same $\hat{p}_{\text{Biden}}$?

Let's explore this in Jupyter!

Sampling distributions

Probability distribution of a statistic

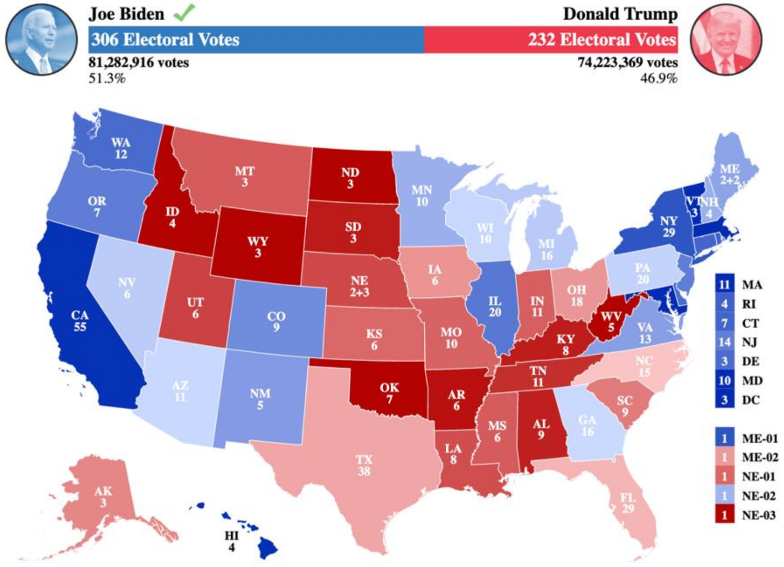
Values of a statistic vary because random samples vary

A **sampling distribution** is a probability distribution of *statistics*

- All possible values of the statistic and all the corresponding probabilities
- We can approximate a sampling distribution by a simulated statistics

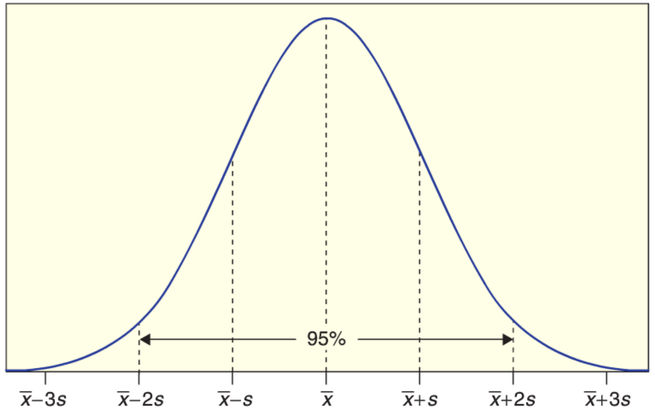
π_{Trump}

$n = 1{,}000$



$\hat{p}_{Trump}$

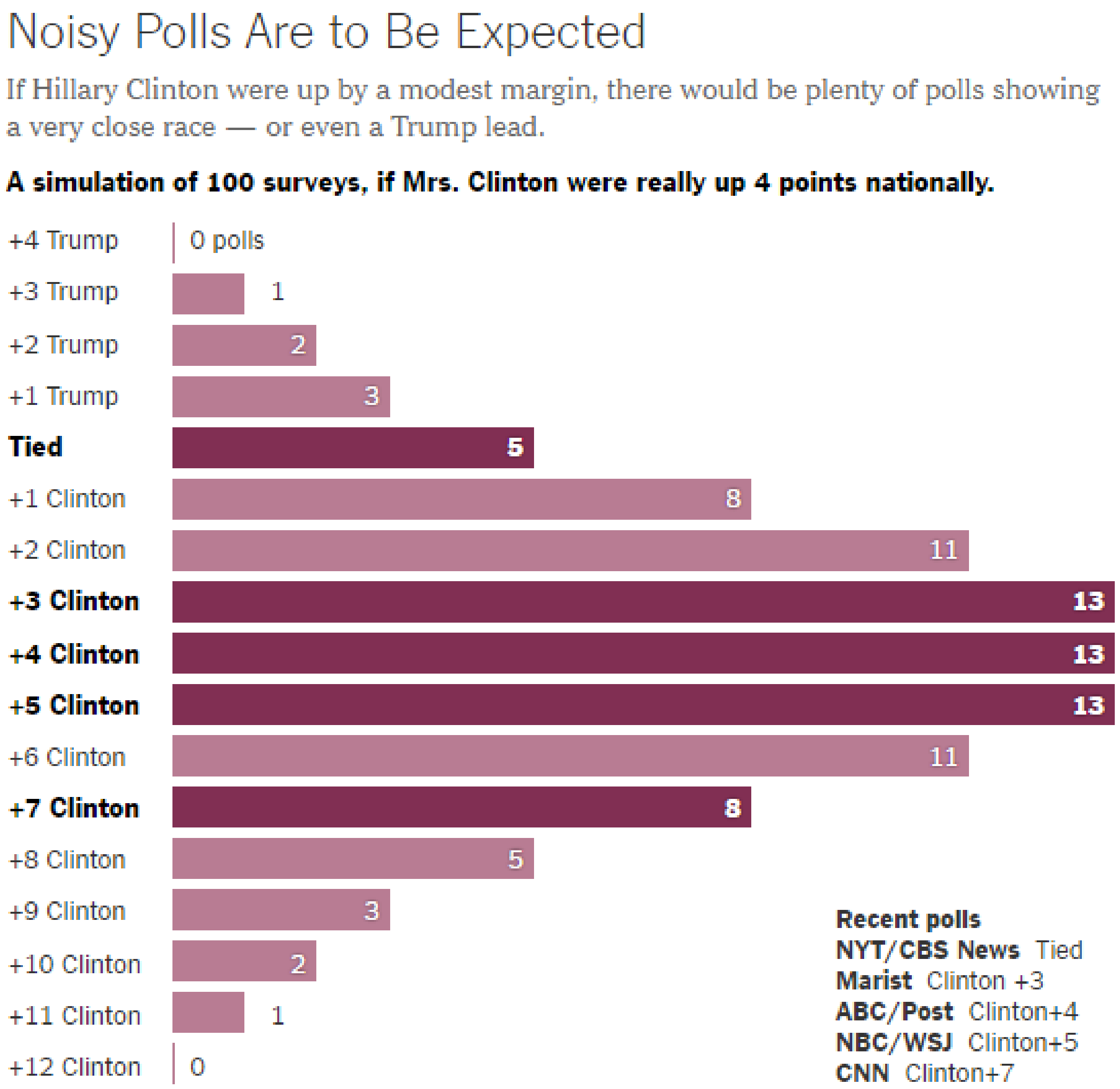
$\hat{p}_{Trump}$



$\hat{p}_{Trump}$

Sampling distribution!

Confused by Contradictory Polls? Take a Step Back



What is this called?

What parameter are they trying to estimate?

NY Times 7/18/16

Let's explore this in Jupyter!