2019 Workshop on Gravitational Waves and High-Performance Computing

Geoffrey Lovelace

August 19, 2019 – August 23, 2019

Day 2

Clicker question #2.3

- What does this program print?

```
j = 1
while j < 3:
    j = j + 1
print(j)
```

A	1
B	2
C	3
D	4

Clicker question #2.4

- What does this program print?

```
product = 1
j = 1
while j < 3:
    product = product * j
    j = j + 1
print(product)
```

A	1
B	2
C	6
D	24

Clicker question #2.4b

- What does this program print?

```
product = 1
j = 1
while j < 4:
    product = product * j
    j = j + 1
print(product)
```

A	1
B	2
C	6
D	24

Clicker question #2.4c

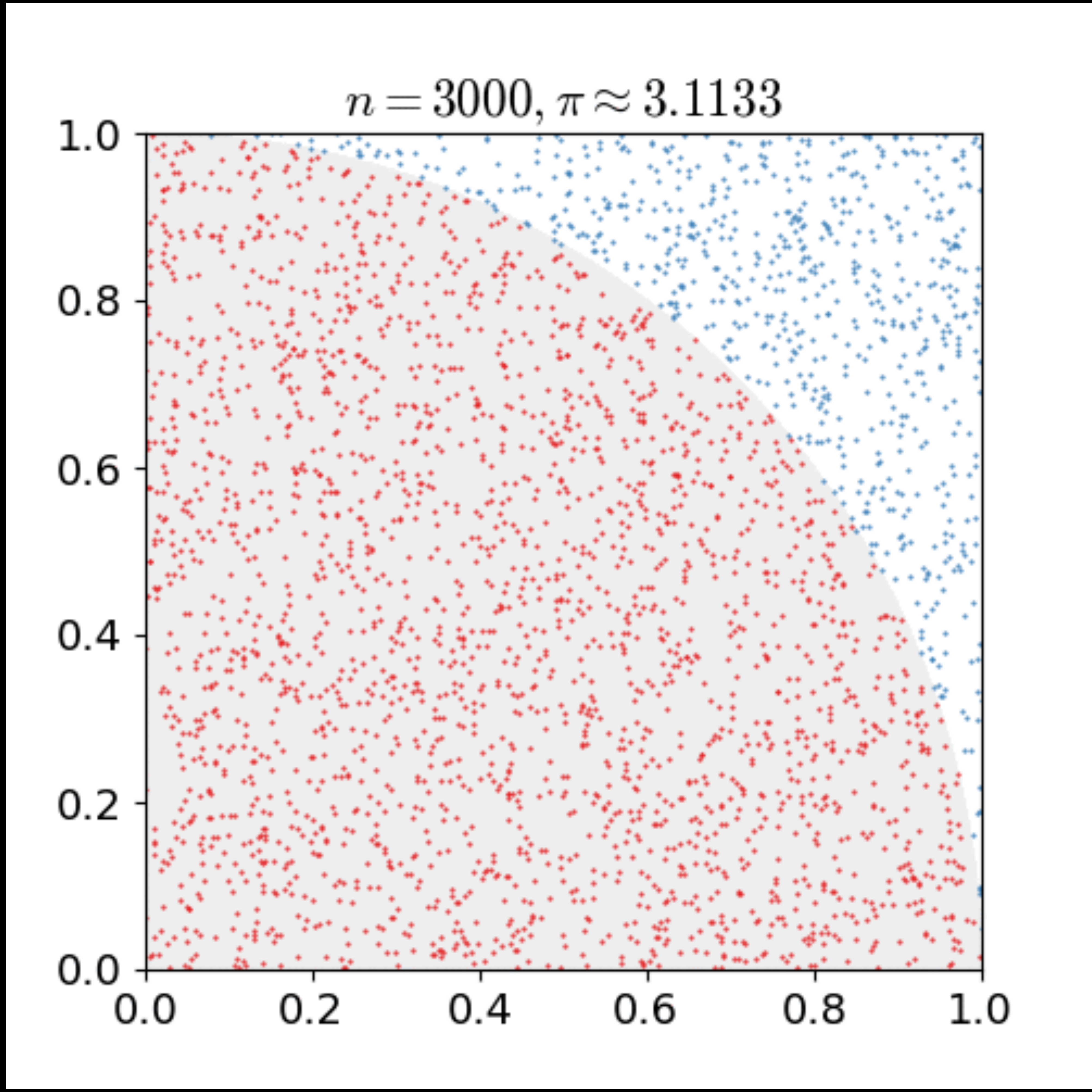
- What value of x makes the program print 24?

```
product = 1
j = 1
while j < x:
    product = product * j
    j = j + 1
print(product)
```

A	3
B	4
C	5
D	6

A silly way to compute π

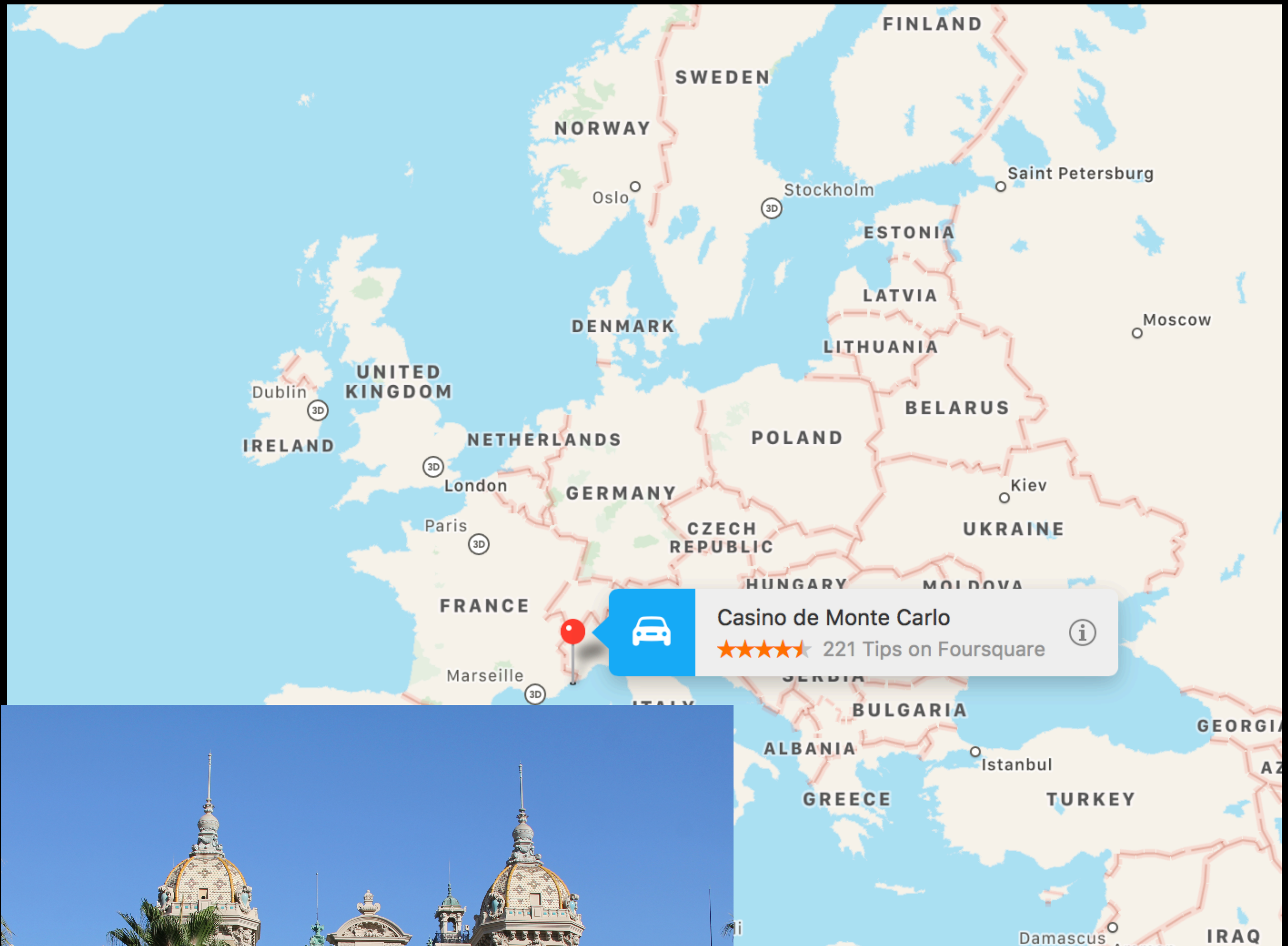
- Throw darts in square
  - (circle area) ÷ (square area) ≈ darts in circle ÷ darts in square = π/4



Courtesy wikipedia

Monte Carlo methods

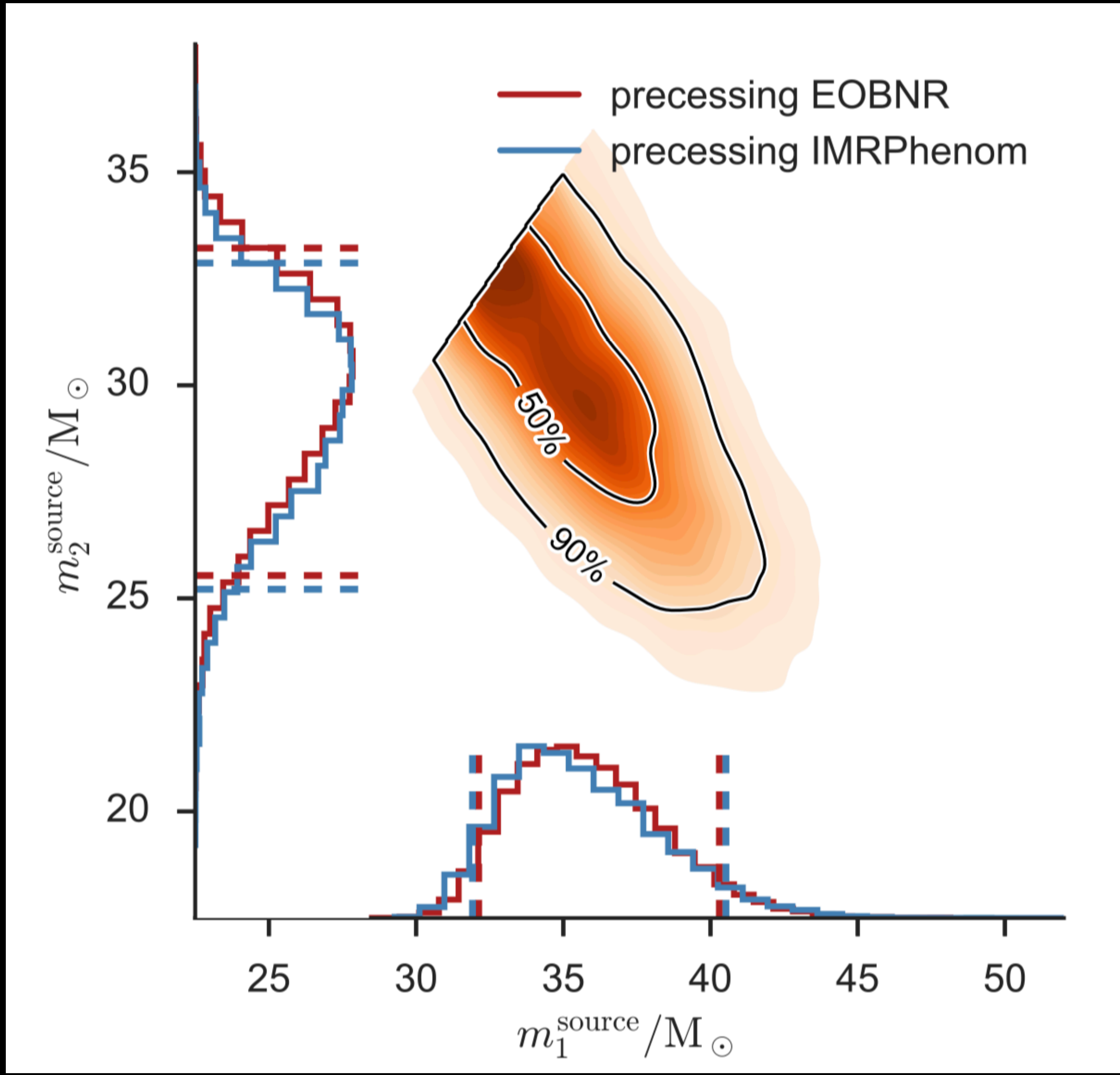
- This idea might seem silly, but it actually has a lot of uses in physics
- **Monte Carlo methods: use repeated random numbers to get results**
  - Min/max of functions
    especially functions of many variables
  - Integrals
    especially high dimensional
  - Explore probability distributions



Images courtesy Wikipedia, Apple Maps

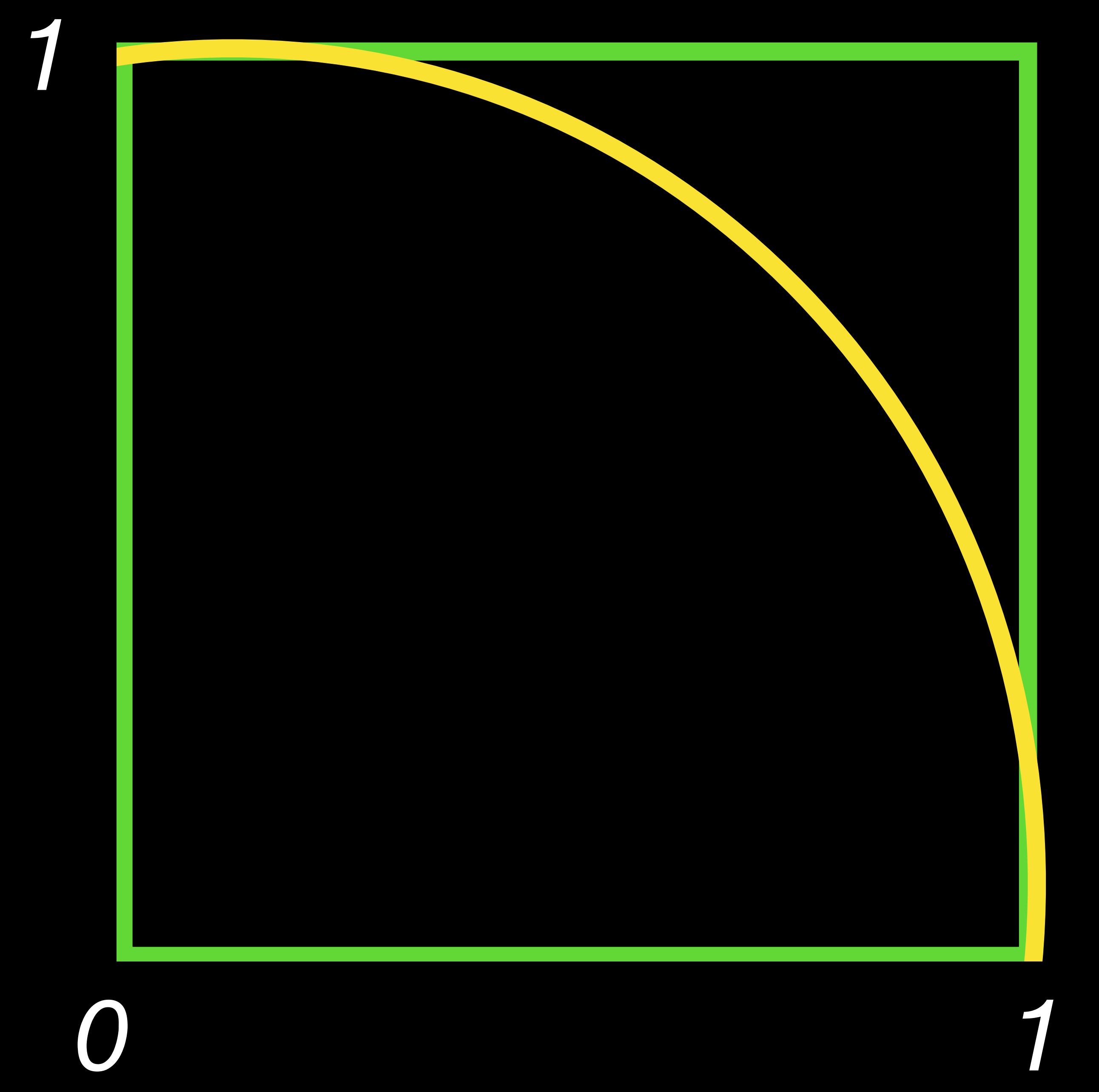
Monte Carlo methods

- This idea might seem silly, but it actually has a lot of uses in physics
- When we observe a gravitational wave from merging black holes…
  - What kinds of black holes made the waves?
    - Choose random parameters (masses, spins, …)
    - Compute the corresponding grav. wave
    - More likely to call the wave a “hit” the better it matches—vs. the last wave “hit”



GW150914: Abbott+ (2016)

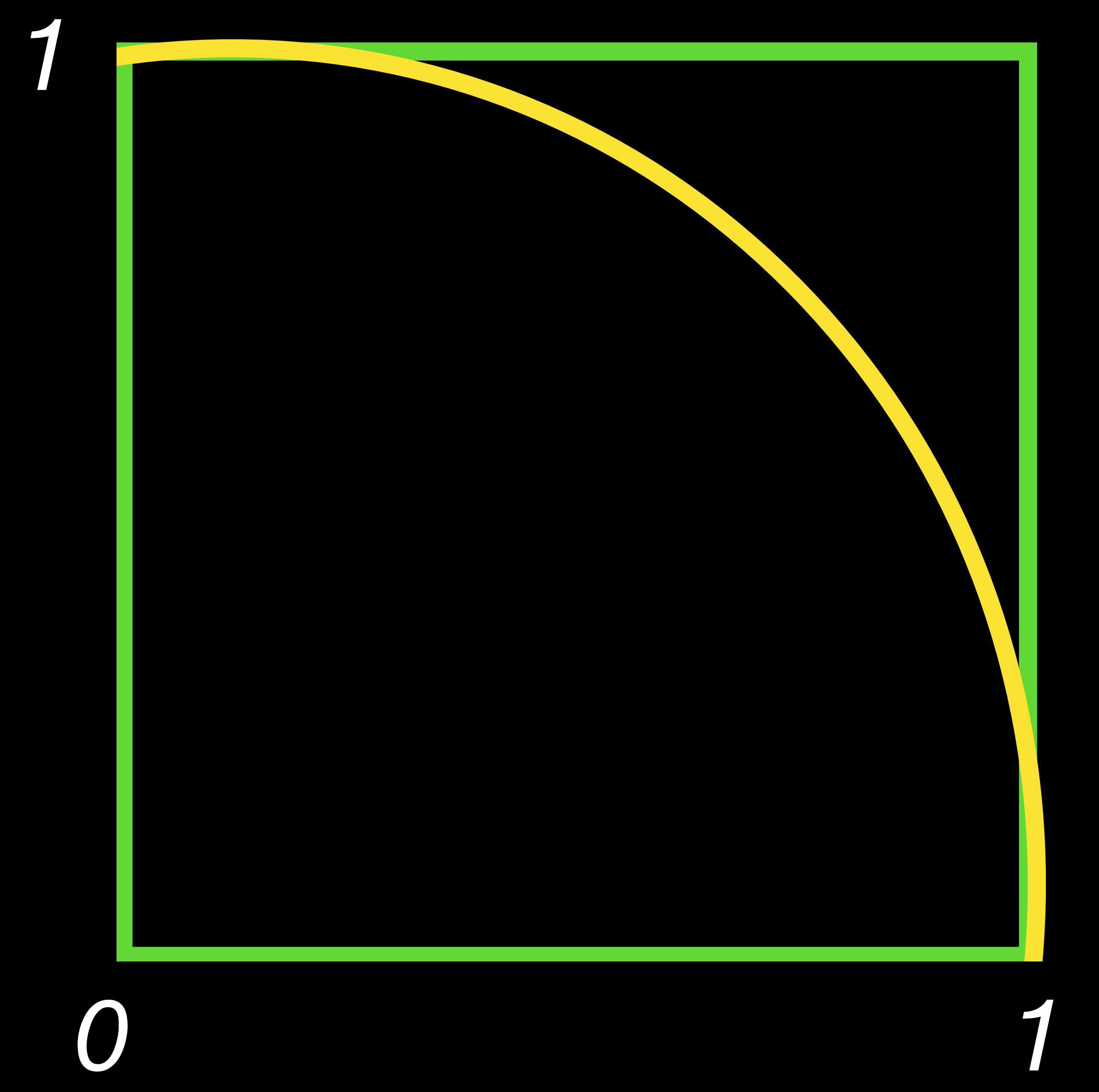
Pi Dartboard 1



- Write a program that prints one random number between 0 and 1

```
import math
import random

print(random.random())
```

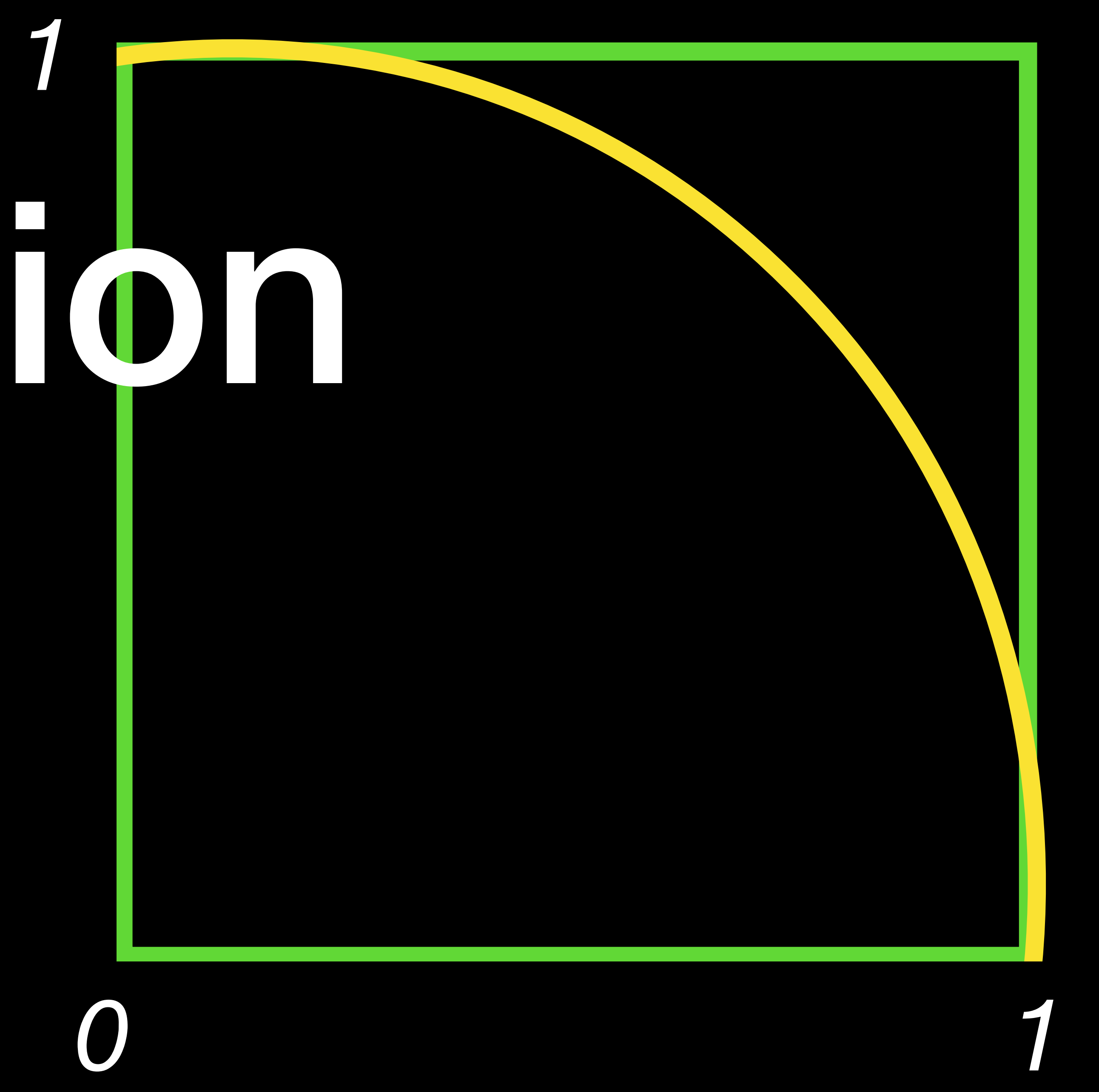
Pi Dartboard 2



- **Challenge:** Modify your program
  - Store the random number in a variable x
  - Store a second random number in a variable y
  - Print x and y

```python
import math
import random
print(random.random())
```

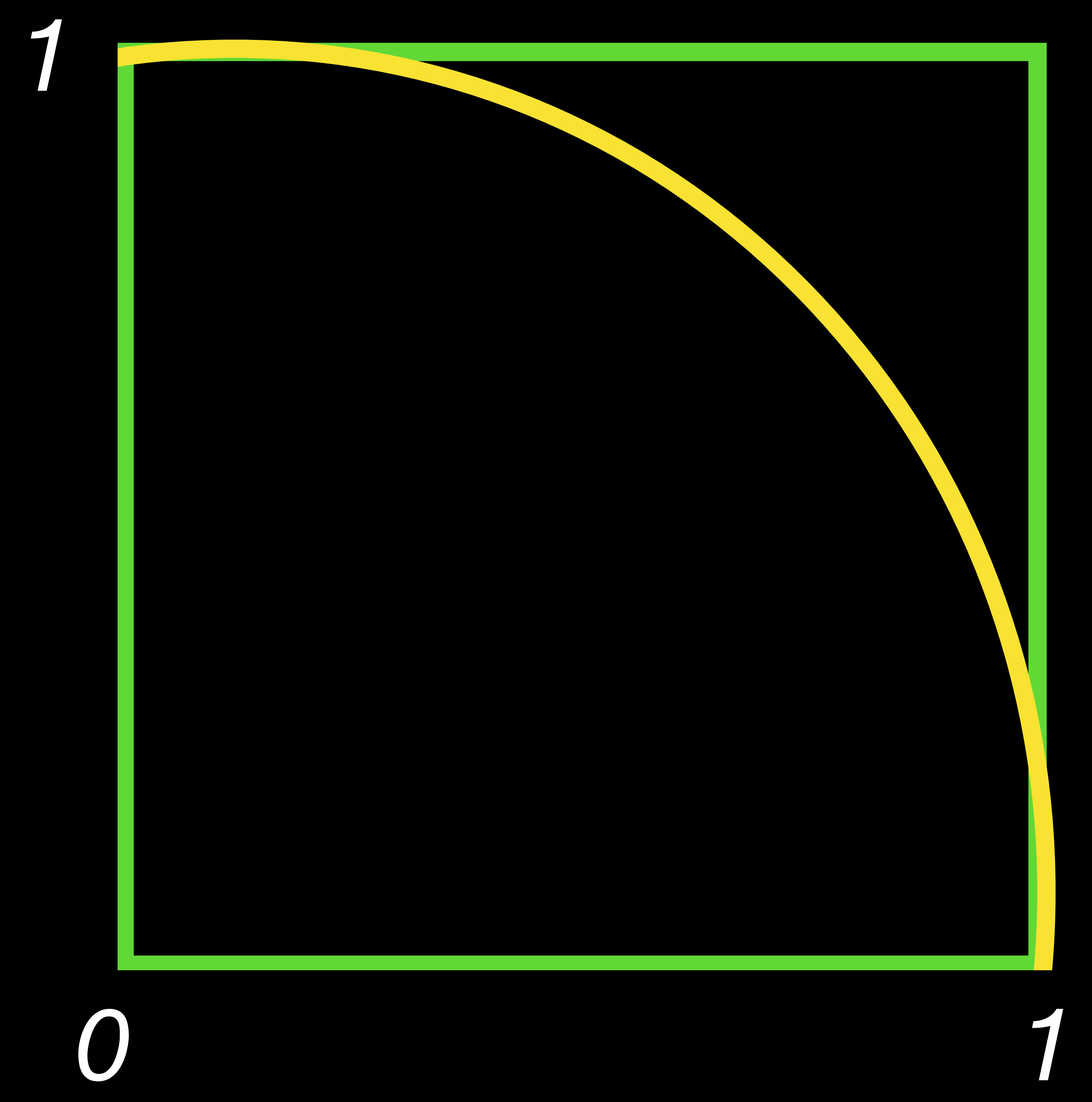
Pi Dartboard 2 Solution



- **Challenge:** Modify your program
  - Store the random number in a variable x
  - Store a second random number in a variable y
  - Print x and y

```
import math
import random

x = random.random()
y = random.random()

print(x)
print(y)
```

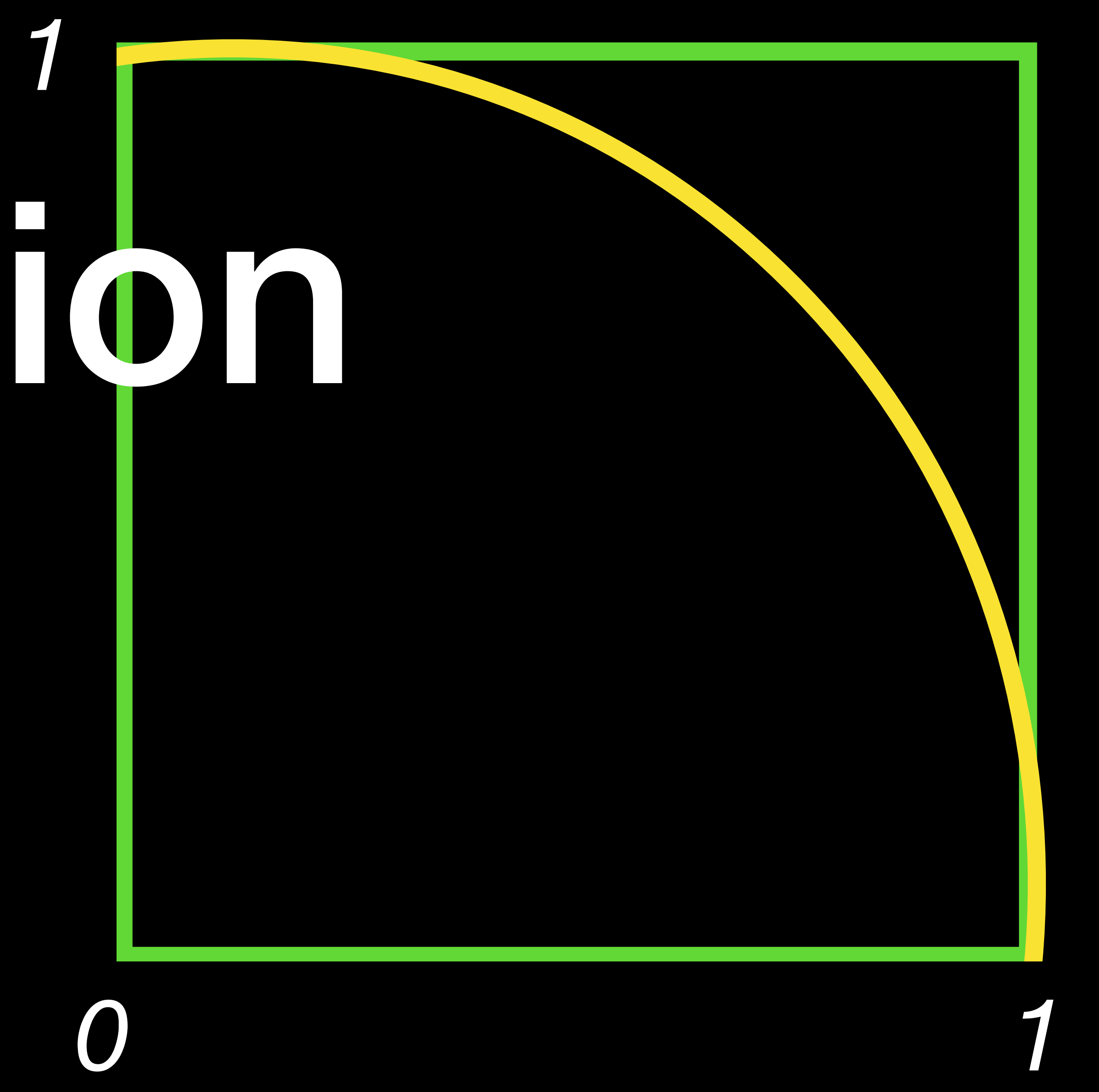
Pi Dartboard 3



- **Challenge:** Modify your program
  - Compute $x^2 + y^2$ and store it in a variable rSquared
  - Print rSquared instead of just x and y

```
import math
import random

x = random.random()
y = random.random()

print(x)
print(y)
```

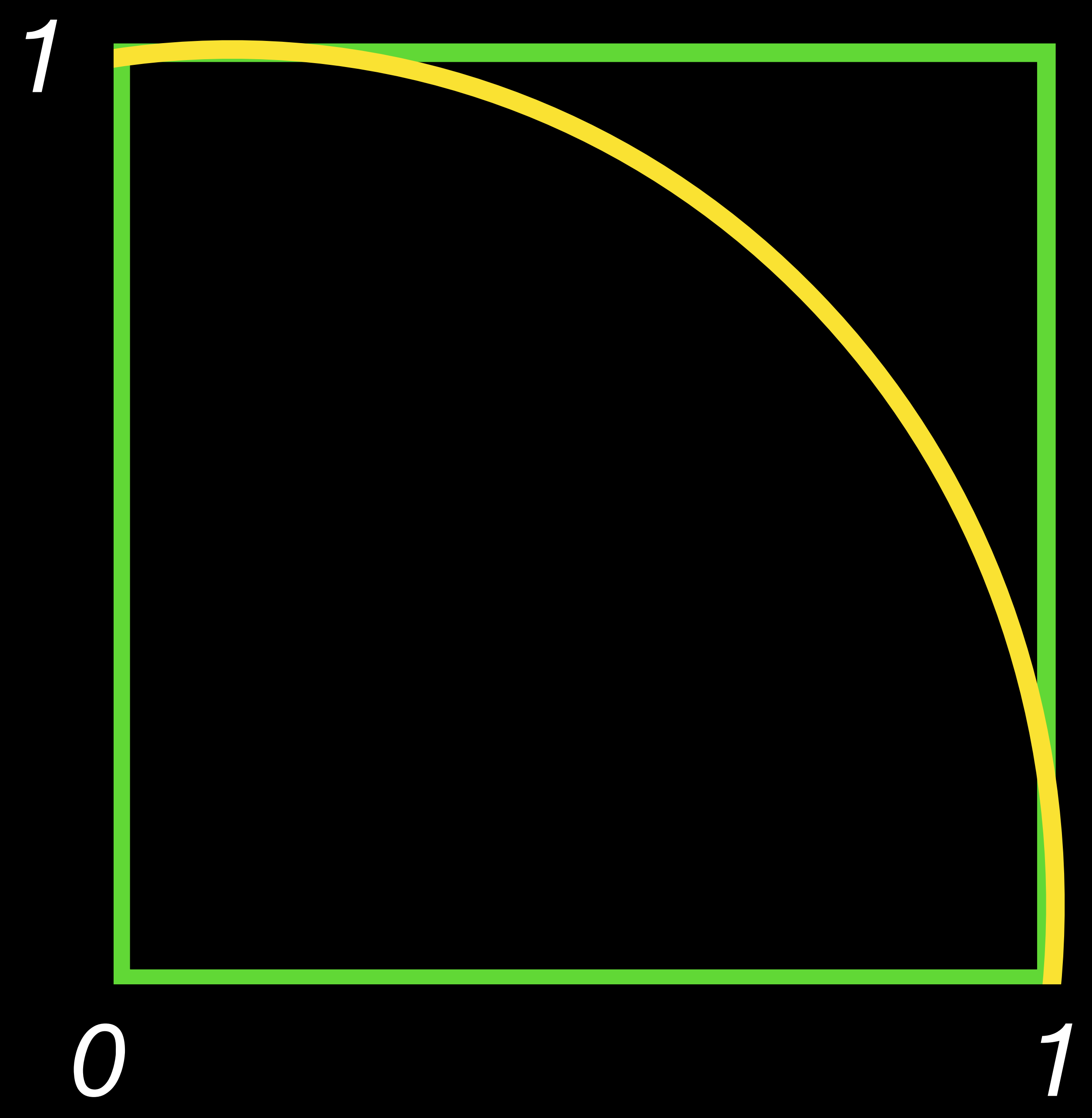
Pi Dartboard 3 Solution



- **Challenge:** Modify your program
  - Compute $x^2 + y^2$ and store it in a variable rSquared
  - Print rSquared instead of just x and y

```
import math
import random

x = random.random()
y = random.random()

rSquared = x**2 + y**2
print(rSquared)
```

Clicker question #2.5

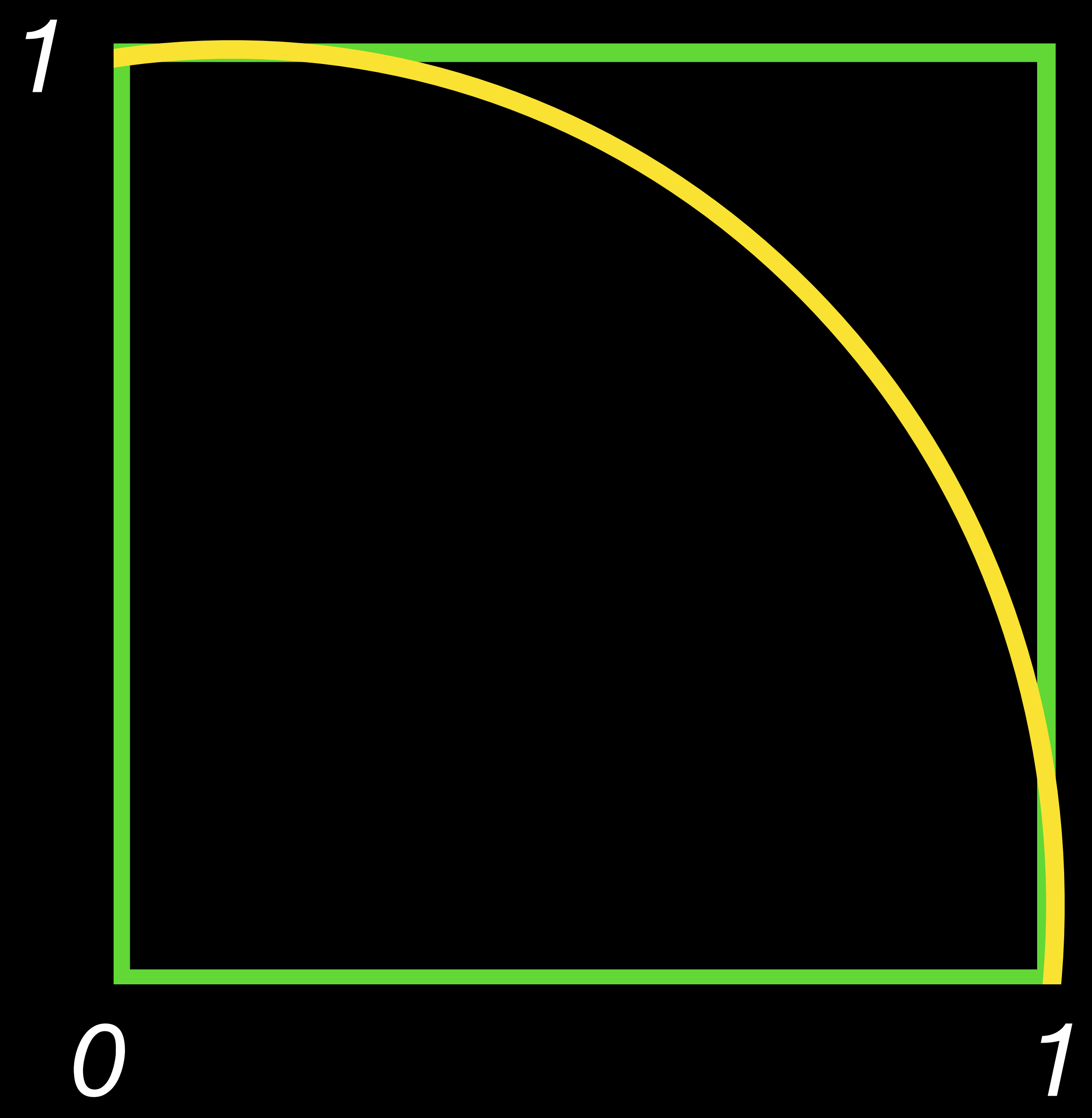


- Which could be a number the program prints?

```python
import math
import random

x = random.random()
y = random.random()

rSquared = x**2 + y**2
print(rSquared)
```

- A -1.51
- B 2.43
- C -0.32
- D 1.01

Clicker question #2.5



- If the dart is inside the **circle**, which could be the number printed by the program?

```python
import math
import random

x = random.random()
y = random.random()

rSquared = x**2 + y**2
print(rSquared)
```

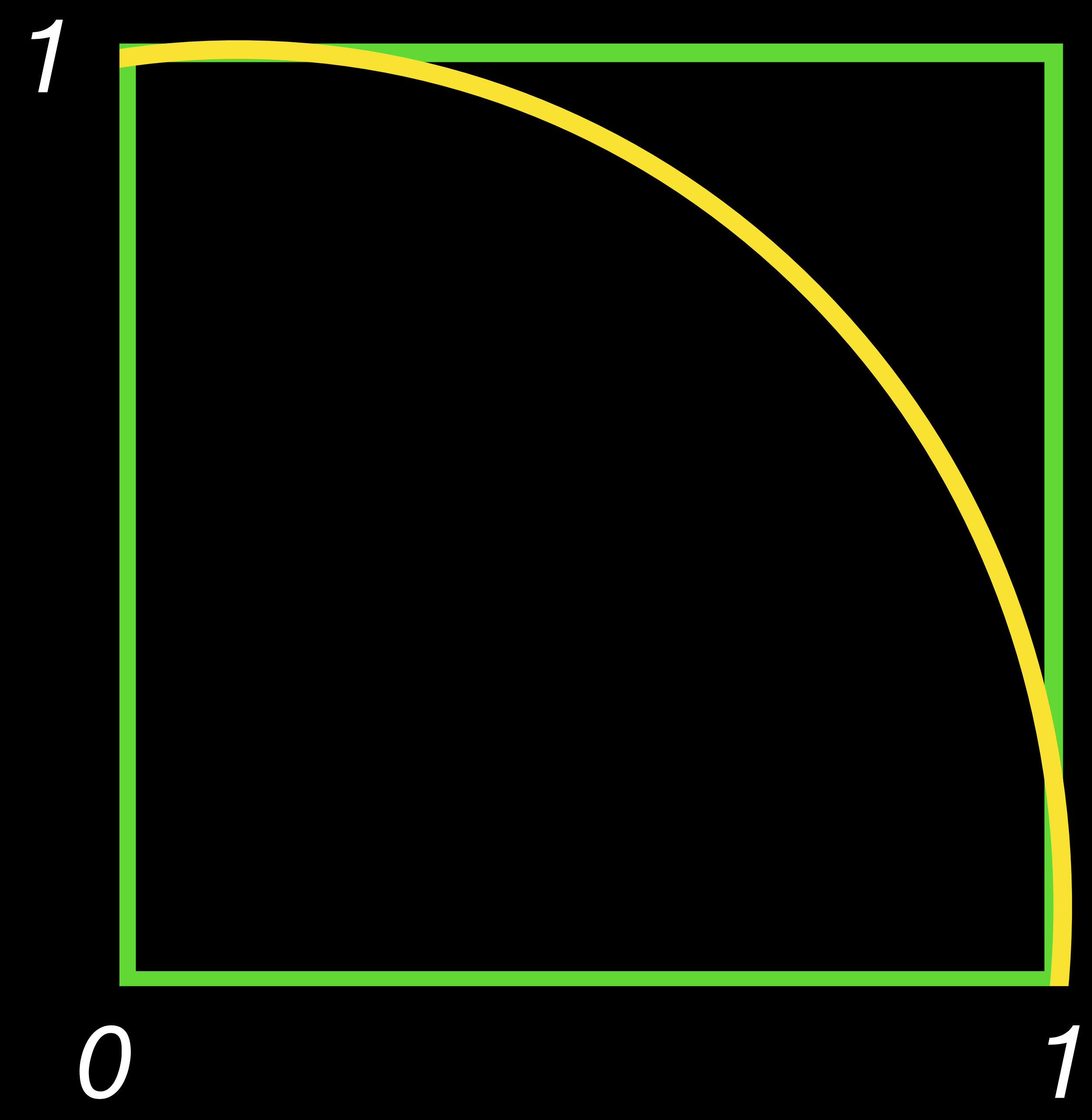
A 1.43

B 0.99

C 1.01

D More than one of ABC

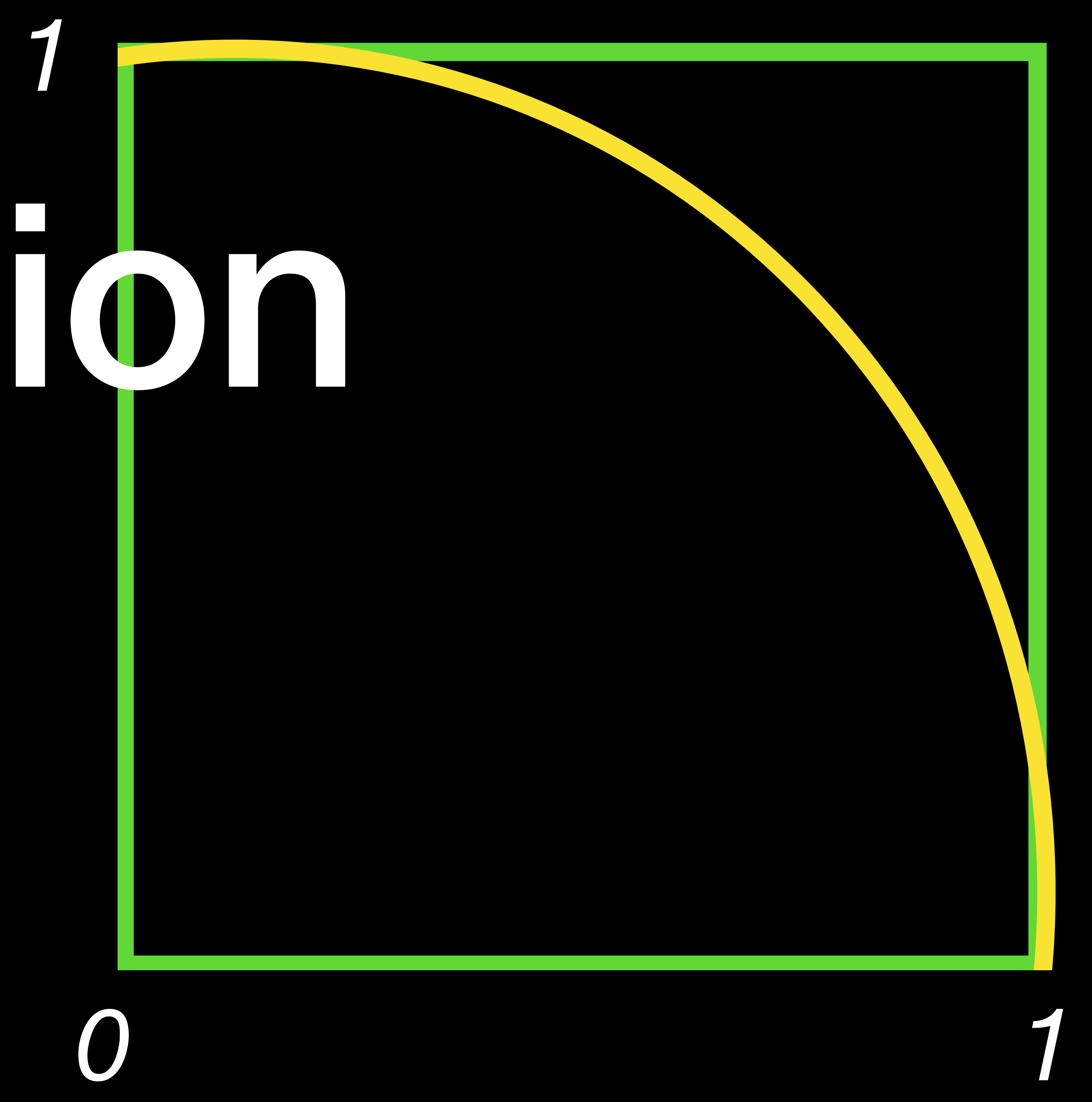
Pi Dartboard 4



- **Challenge:** Modify your program
  - Just below import random, make a new variable called “hits”, set to 0
  - If rSquared < 1, add 1 to hits
  - Print hits instead of rSquared

```
import math
import random
x = random.random()
y = random.random()
rSquared = x**2 + y**2
print(rSquared)
```

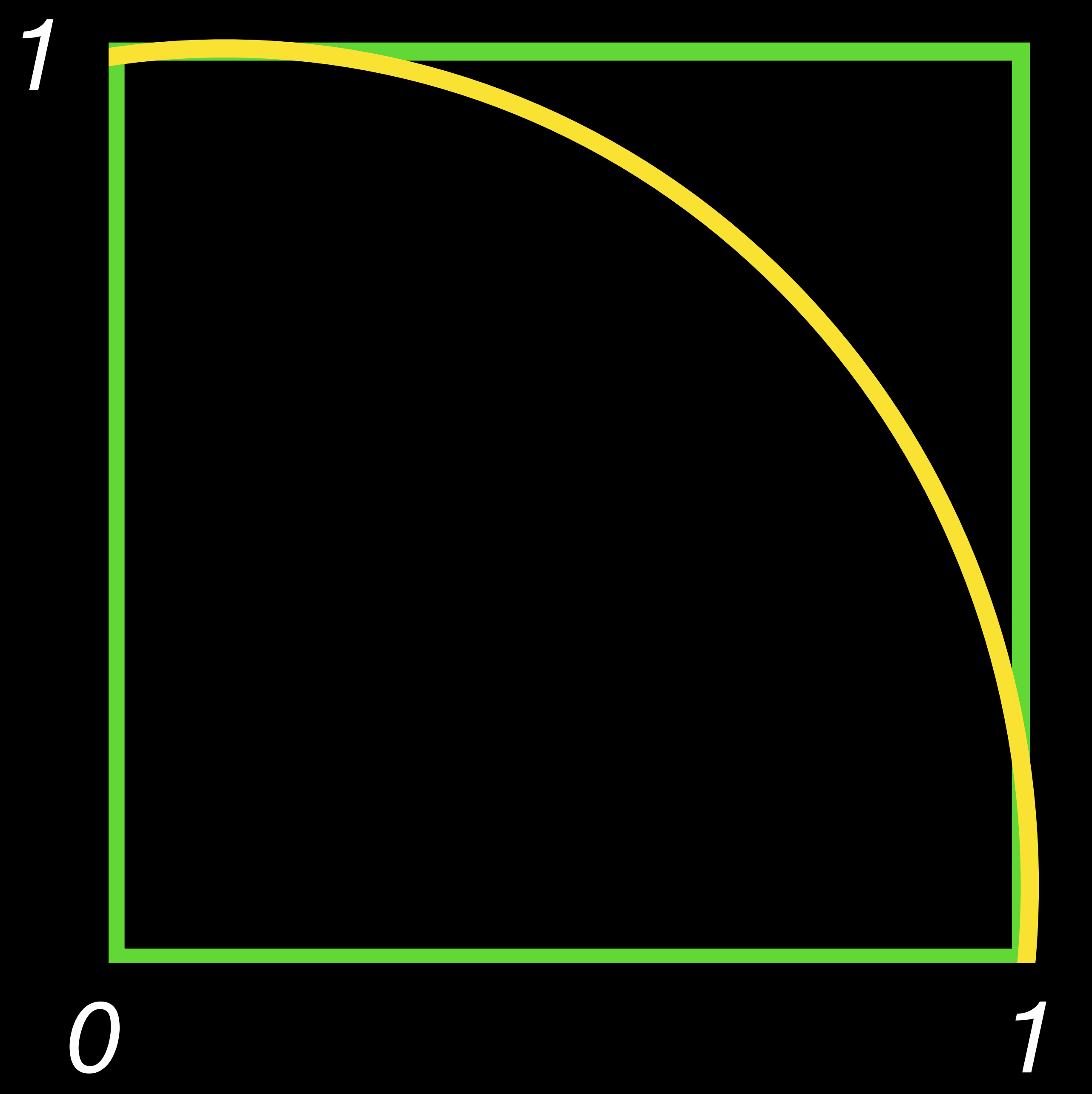
Pi Dartboard 4 Solution



- **Challenge:** Modify your program
  - Just below import random, make a new variable called "hits", set to 0
  - If rSquared < 1, add 1 to hits
  - Print hits instead of rSquared

```
import math
import random
hits = 0
x = random.random()
y = random.random()
rSquared = x**2 + y**2
if rSquared < 1:
    hits = hits + 1
print(hits)
```

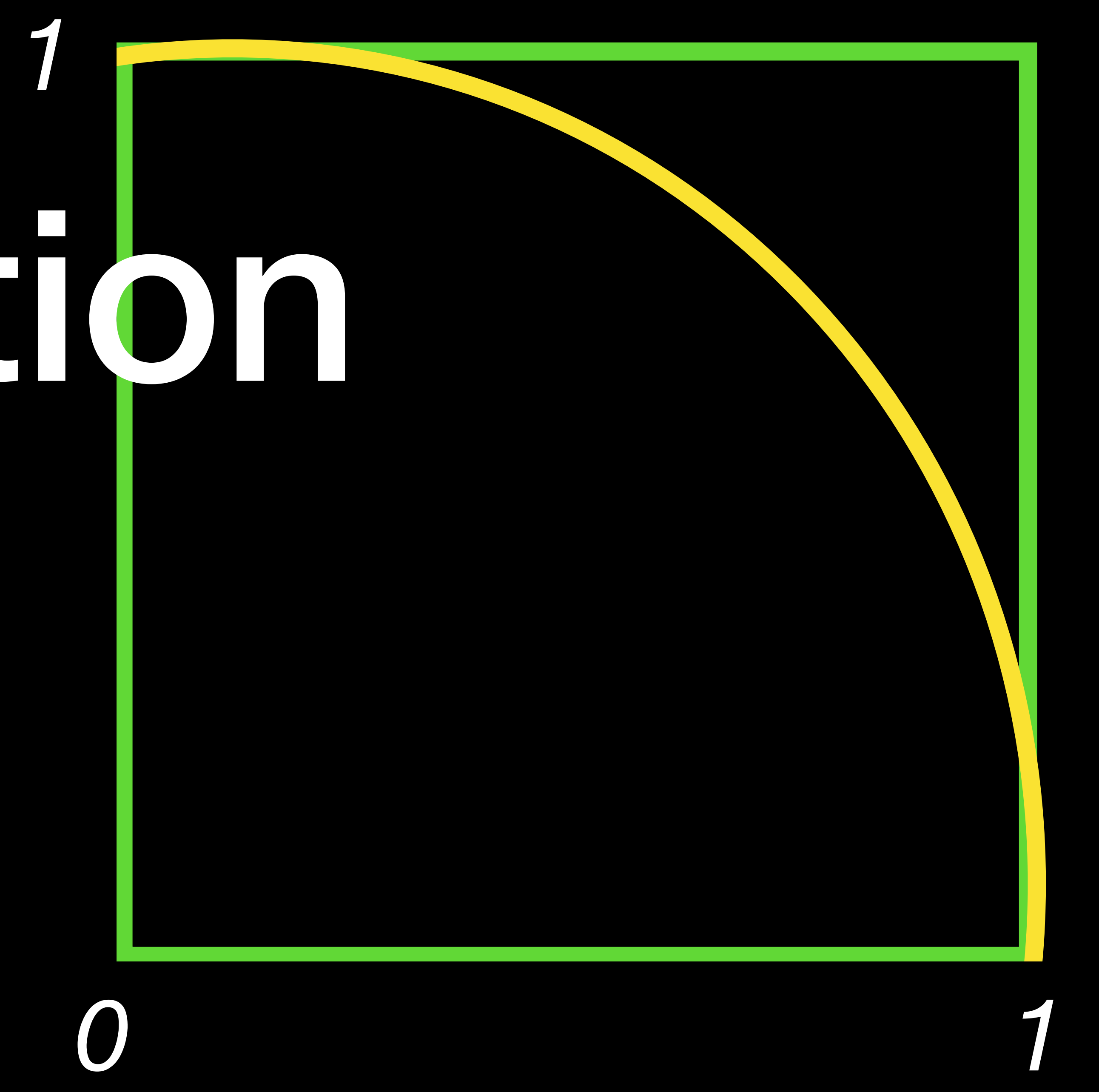
Pi Dartboard 4.5



- **Challenge:** Modify your program
  - Add a new variable, just below hits, called throws. Set it equal to 10.
  - Add a while loop just below your new variable, throws.
    - Make a counter variable (i or j or ...) and set it equal to zero. Then make a while loop. While your counter variable is less than throws, each time through the while loop, add 1 to your counter variable.

```
import math
import random

hits = 0

x = random.random()
y = random.random()

rSquared = x**2 + y**2
if rSquared < 1:
    hits = hits + 1
print(hits)
```

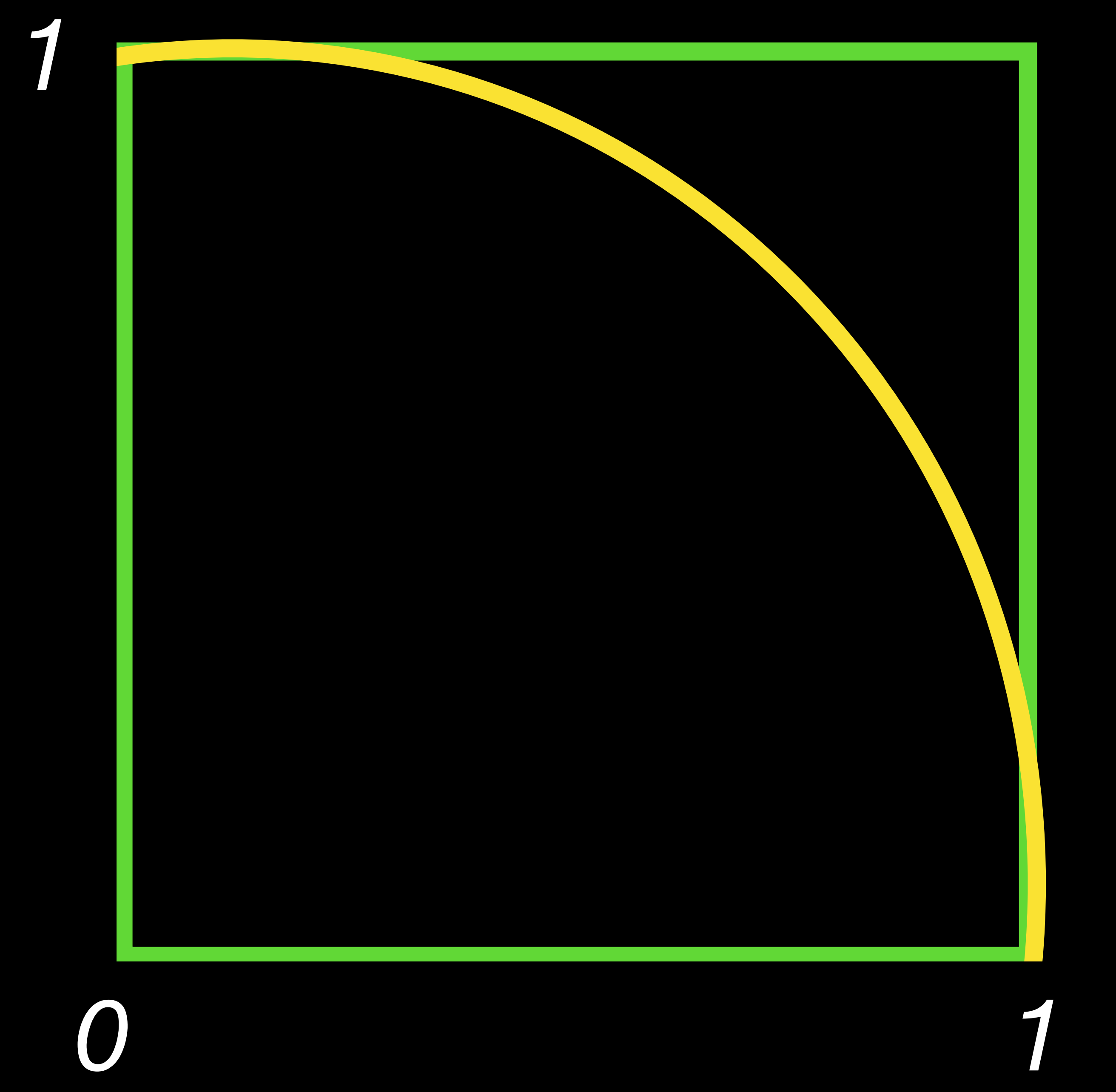
Pi Dartboard 4.5 solution



- **Challenge:** Modify your program
  - Add a new variable, just below hits, called throws. Set it equal to 10.
  - Add a while loop just below your new variable, throws.
    - Make a counter variable (i or j or ...) and set it equal to zero. Then make a while loop. While your counter variable is less than throws, each time through the while loop, add 1 to your counter variable.

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    i = i + 1

x = random.random()
y = random.random()

rSquared = x**2 + y**2
if rSquared < 1:
    hits = hits + 1
print(hits)
```

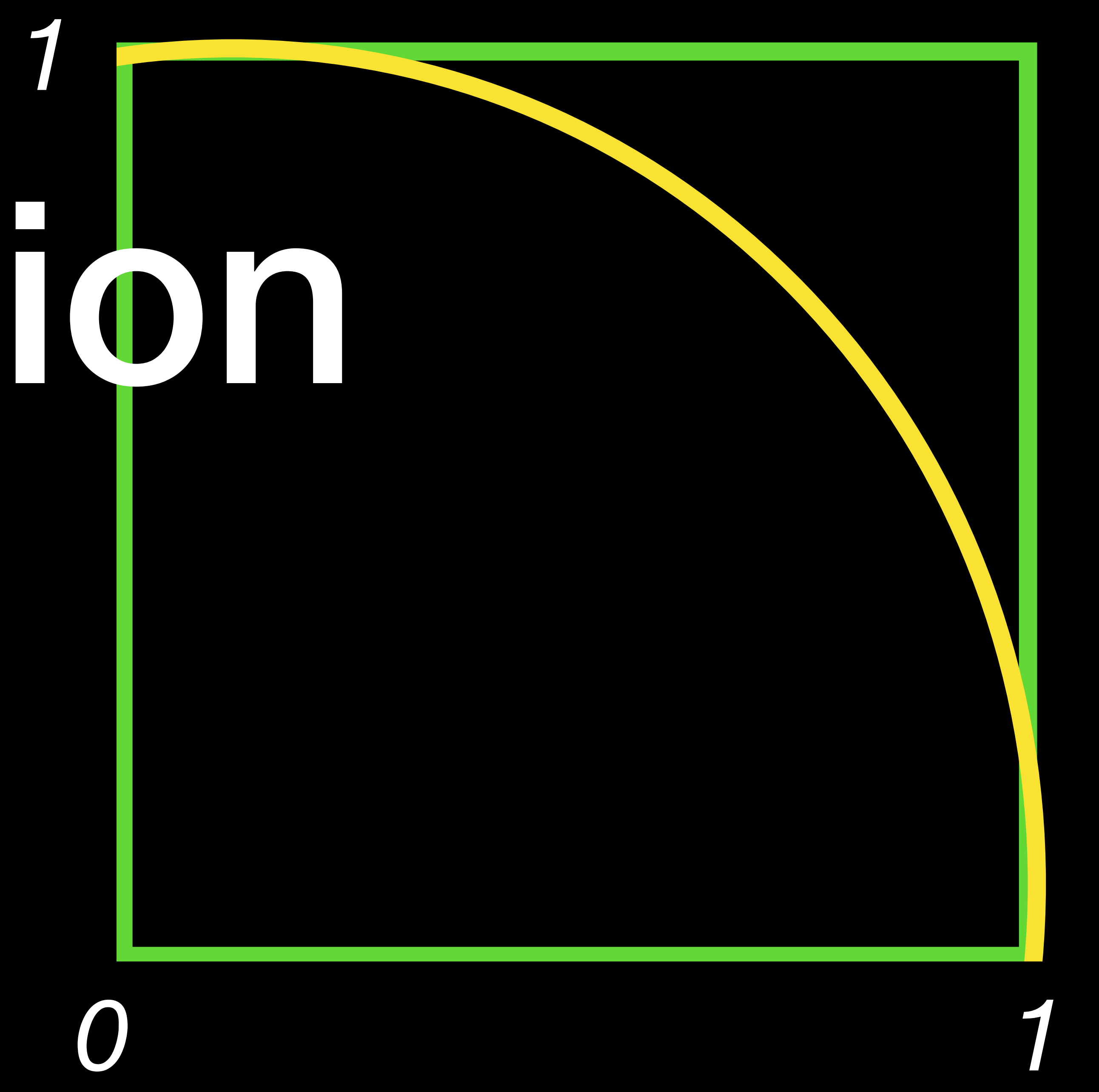
Pi Dartboard 5



- **Challenge:** Modify your program
  - Put the code that throws the dart and sees if it hit inside a while loop, so that you throw 10 darts instead of 1 dart
  - Your print statement should still be outside the while loop

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    i = i + 1

x = random.random()
y = random.random()

rSquared = x**2 + y**2
if rSquared < 1:
    hits = hits + 1
print(hits)
```

Pi Dartboard 5 Solution

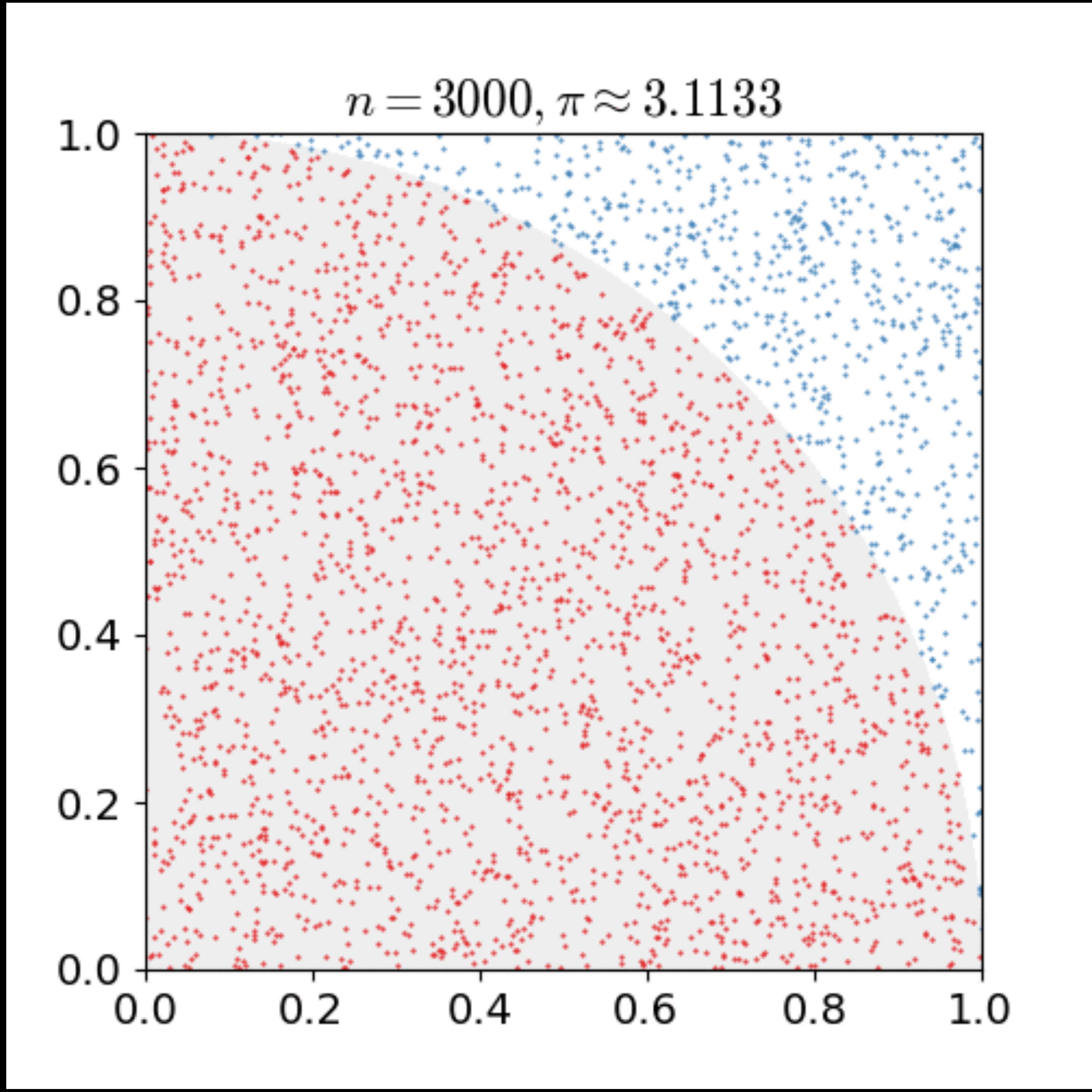


- **Challenge:** Modify your program
  - Add a new variable, just below hits, called throws. Set it equal to 10.
  - Put the code that throws the dart and sees if it hit inside a while loop, so that you throw 10 darts instead of 1 dart

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1

print(hits)
```

A silly way to compute π

- Throw darts in square
  - (circle area) ÷ (square area) ≈ hits ÷ throws = π/4
  - So π ≈ 4 * (hits ÷ throws)

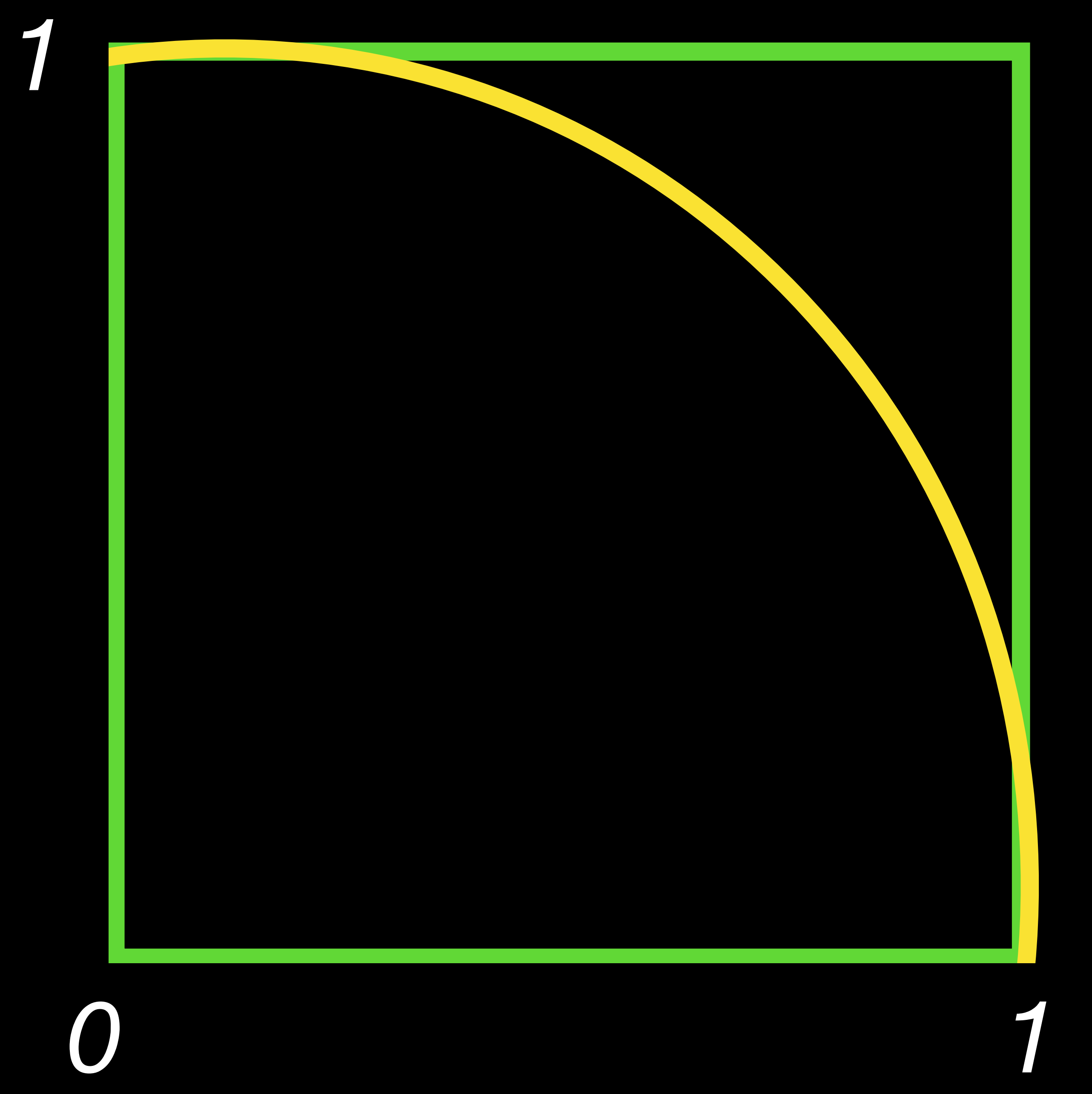


Courtesy wikipedia

Pi Dartboard 6

- Finish the dartboard
  - Make a variable pi, set to 4.0 * float(hits) / float(throws)
    - Should you do this inside or outside the while loop?
  - Print pi

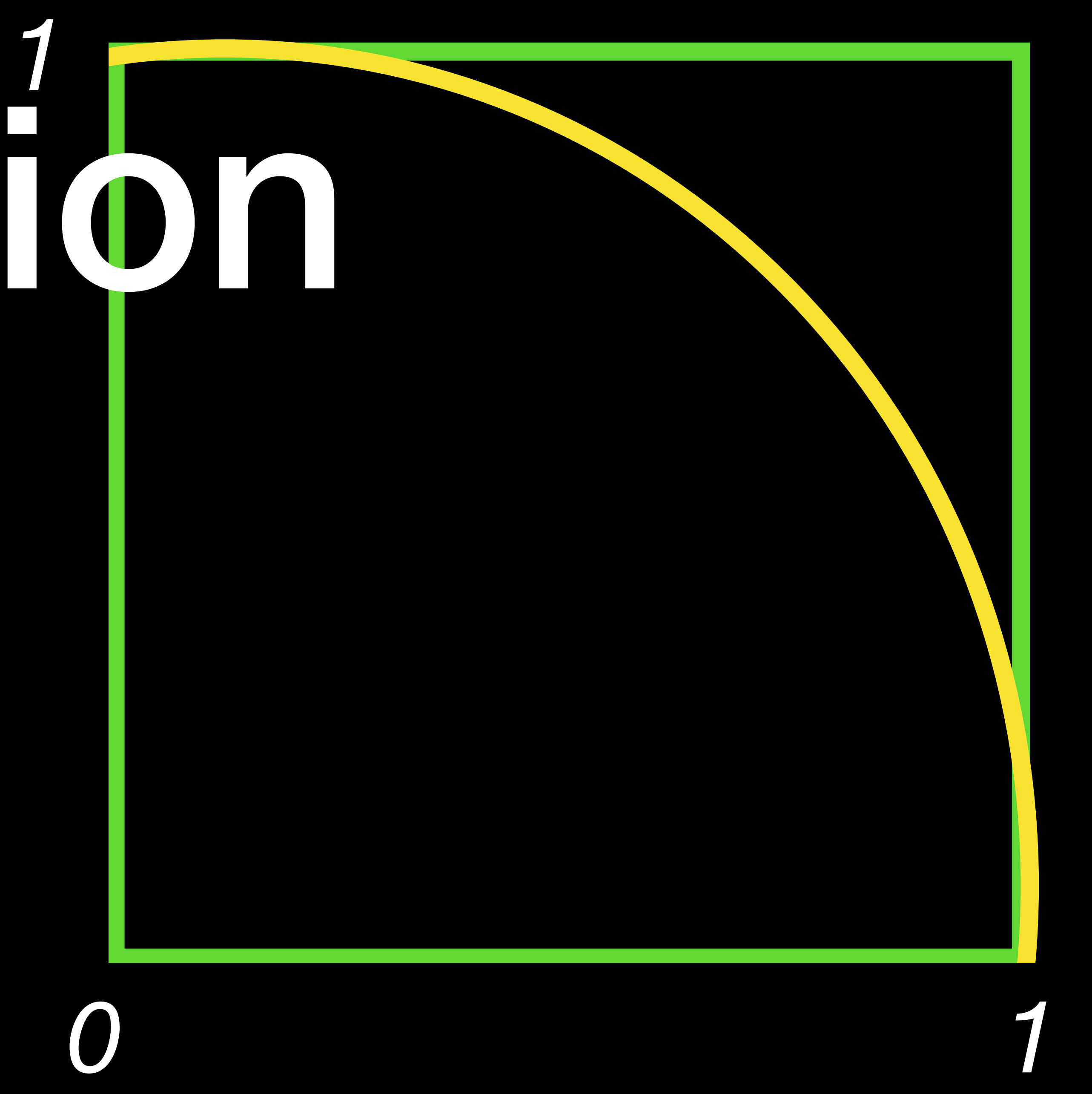
```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1

print(hits)
```



Pi Dartboard 6 Solution

- Finish the dartboard
  - Compute pi as 4.0 * float(hits) / float(throws)
  - Print your pi estimate

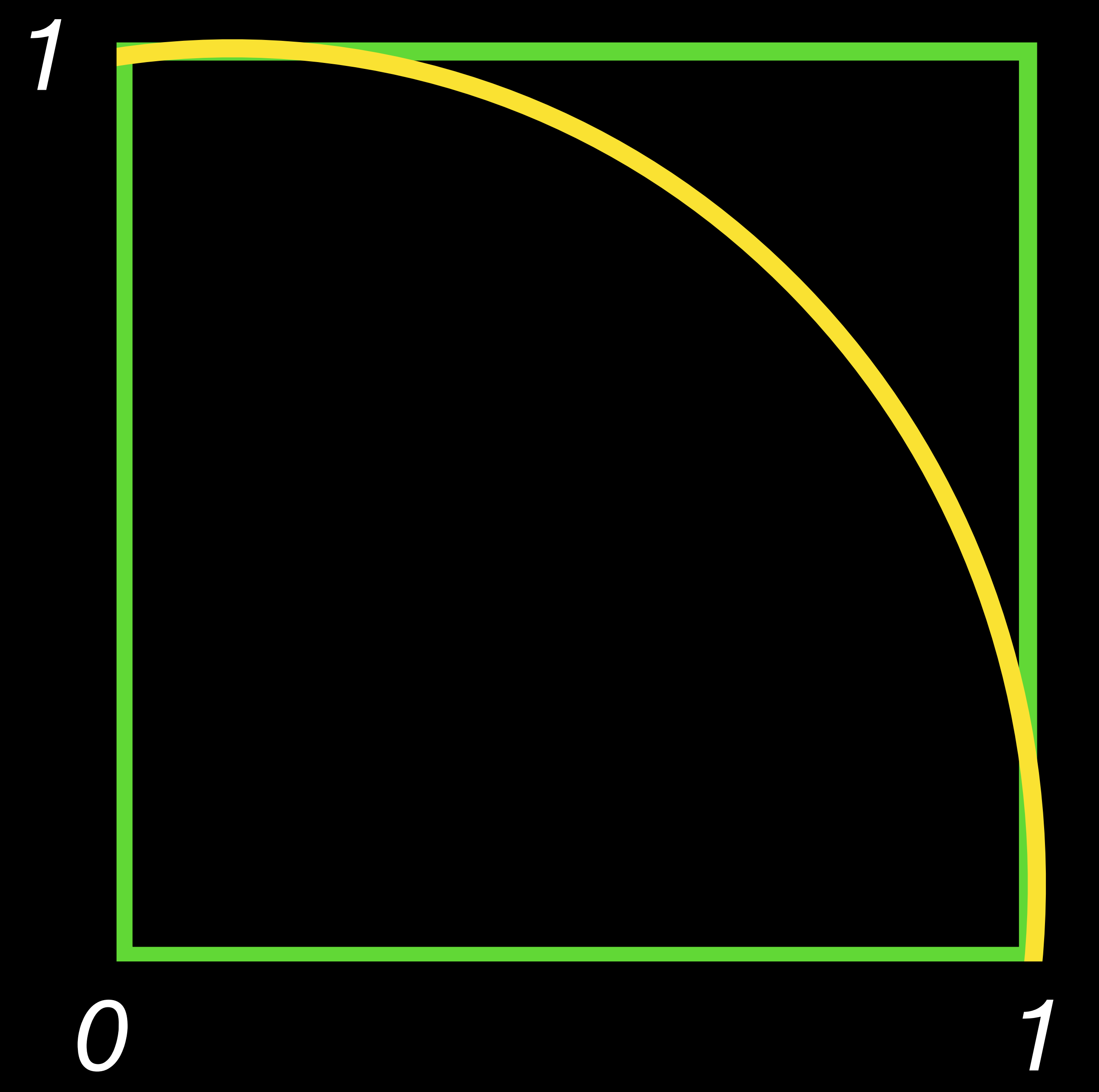
```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1

pi = 4.0 * float(hits) / float(throws)
print(pi)
```



Pi Dartboard 7

- The tutor won't let us run lots of darts
- So paste this into a cell in Jupyter on colab.google.com and run it
  - See what happens as you make throws 10**n, n=1,2,3,4,5,6,7

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1

pi = 4.0 * float(hits) / float(throws)
print(pi)
```



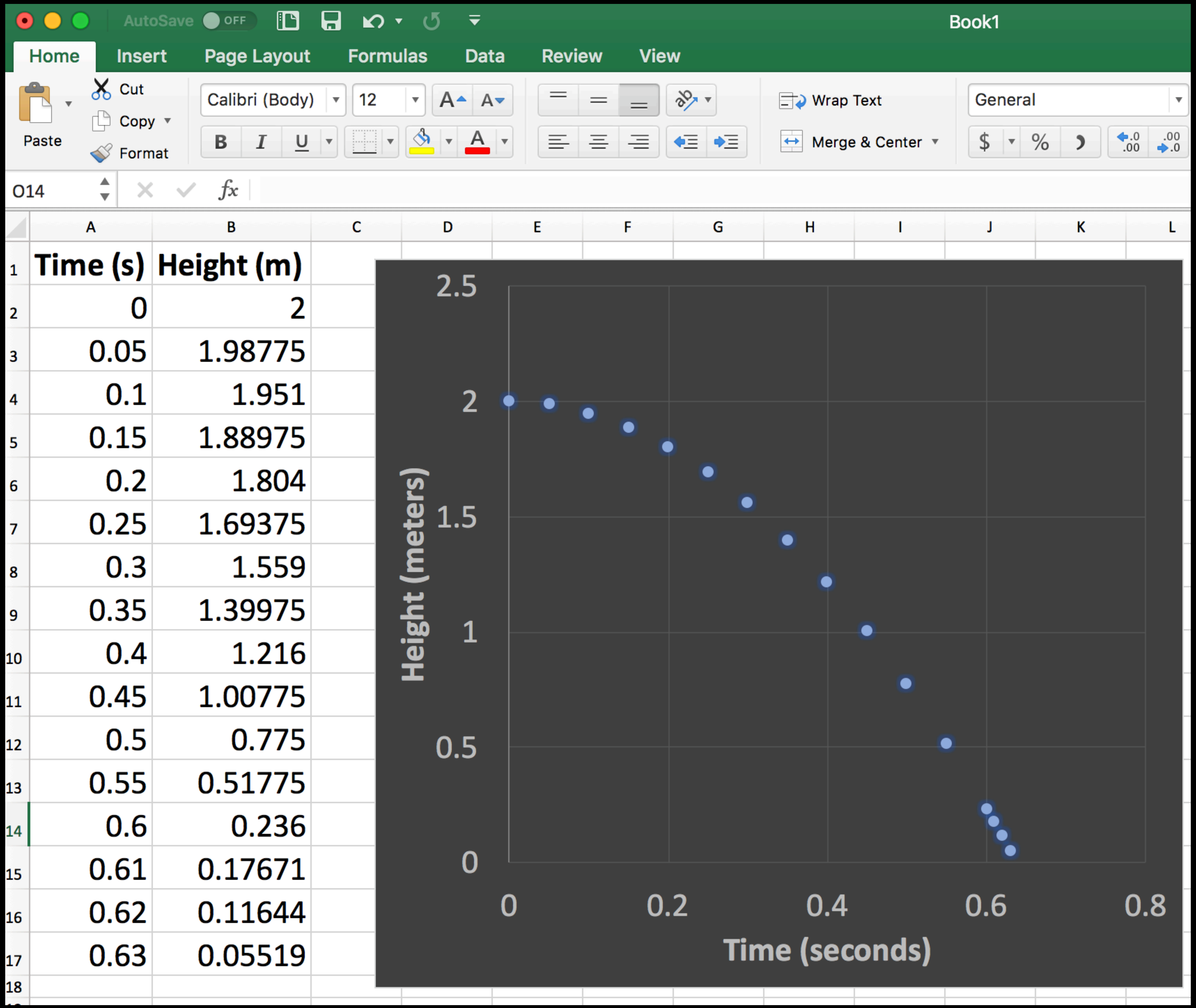
- https://mybinder.org/v2/gh/geoffrey4444/NRDataExample/master

Plotting your results

- Scatter plots
- Lists and numpy arrays
- Pyplot plotting

Scatter plots

- Data: result or output given some input
- Example: dropped marker height vs. time
- Tools to make scatter plots
  - Excel
  - Python
    - Lists of numbers
    - Computations on lists of numbers: numpy arrays
    - pyplot: makes scatter plots



Time (s)	Height (m)
0	2
0.05	1.98775
0.1	1.951
0.15	1.88975
0.2	1.804
0.25	1.69375
0.3	1.559
0.35	1.39975
0.4	1.216
0.45	1.00775
0.5	0.775
0.55	0.51775
0.6	0.236
0.61	0.17671
0.62	0.11644
0.63	0.05519

Lists

- **Values -** ordered sets of objects, all the same type (like floats or ints)
- **Operators - [ ], .append()**
- **Easily add on elements in loops** with .append()

```
x = [-2.0, -1.0, 0.0, 1.0, 2.0]
y = ["Hello", "world"]
z = [1, 4, 9, 16]
```

```
z.append(25) == [1, 4, 9, 16, 25]
```

```
x[0] == -2.0
x[1] == -1.0
x[4] == 2.0
y[0] == "Hello"
y[-1] == "world"
z[-1] == 16
z[0] == 1
```

Loop over lists

```
for i in [1,2,3,4]:
  print(i*i)
```

```
0
1
4
9
```

```
import numpy as np
print(np.arange(1,5,1))
[1,2,3,4]
```

```
import numpy as np
myCountArray = np.arange(1,5)
myList = []
for i in myCountArray:
    myList.append(i*i)
print(myCountArray)
print(myList)
```

```
[1,2,3,4]
[1,4,9,16]
```

Clicker question #2.6

- What value does the program print?

```
x = [1.0, 4.0, 9.0]
print(x[1])
```

A	1.0
B	4.0
C	9.0
D	The entire list

Numpy arrays

- **Values -** ordered sets of objects, all the same type (like floats or ints)
- **Operators - [ ], +, -, *, /, np.sqrt(), np.sin(), np.cos(), …**
- **Easily do math on whole lists at once** (like formulas in Excel)

```
x = np.array([-2.0, -1.0, 0.0, 1.0, 2.0])
y = np.array(["Hello", "world"])
q = np.array([1, 2, 3, 4])
r = q * 2
s = q + r
z = q * q
```

```
r == np.array([2, 4, 6, 8])
```

```
s == np.array([3, 6, 9, 12])
z == np.array([1, 4, 9, 16])
```

```
x[0] == -2.0
x[1] == -1.0
x[4] == 2.0
y[0] == "Hello"
y[-1] == "world"
z[-1] == 16
z[0] == 1
```

Making sample data

- Annoying to type [1,2,3,4,…] all the time
- Instead: np.arange(start, stop, step)
- What do all these numbers mean??
    - Make a plot to visualize them

Try in colab!

```python
import numpy as np
x = np.arange(-4.0, 4.0, 0.01)
y = np.sin(x)**3
print(x)
print(y)
```

```
-9.99825171e-01 -9.99351433e-01 -9.98578166e-01 -9.97505912e-01
-9.96135421e-01 -9.94467651e-01 -9.92503769e-01 -9.90245148e-01
-9.87693366e-01 -9.84850205e-01 -9.81717651e-01 -9.78297888e-01
-9.74593301e-01 -9.70606471e-01 -9.66340175e-01 -9.61797379e-01
-9.56981241e-01 -9.51895105e-01 -9.46542499e-01 -9.40927131e-01
-9.35052889e-01 -9.28923832e-01 -9.22544191e-01 -9.15918365e-01
-9.09050915e-01 -9.01946561e-01 -8.94610179e-01 -8.87046794e-01
-8.79261581e-01 -8.71259853e-01 -8.63047062e-01 -8.54628794e-01
-8.46010761e-01 -8.37198799e-01 -8.28198860e-01 -8.19017011e-01
-8.09659425e-01 -8.00132377e-01 -7.90442239e-01 -7.80595473e-01
-7.70598629e-01 -7.60458333e-01 -7.50181290e-01 -7.39774268e-01
-7.29244102e-01 -7.18597680e-01 -7.07841944e-01 -6.96983877e-01
-6.86030504e-01 -6.74988880e-01 -6.63866088e-01 -6.52669231e-01
-6.41405427e-01 -6.30081800e-01 -6.18705479e-01 -6.07283586e-01
-5.95823237e-01 -5.84331527e-01 -5.72815532e-01 -5.61282298e-01
-5.49738839e-01 -5.38192126e-01 -5.26649084e-01 -5.15116589e-01
-5.03601455e-01 -4.92110435e-01 -4.80650212e-01 -4.69227393e-01
-4.57848505e-01 -4.46519990e-01 -4.35248195e-01 -4.24039375e-01
-4.12899678e-01 -4.01835147e-01 -3.90851715e-01 -3.79955193e-01
-3.69151273e-01 -3.58445520e-01 -3.47843366e-01 -3.37350109e-01
-3.26970907e-01 -3.16710771e-01 -3.06574566e-01 -2.96567003e-01
-2.86692639e-01 -2.76955868e-01 -2.67360924e-01 -2.57911871e-01
```

Plotting sample data

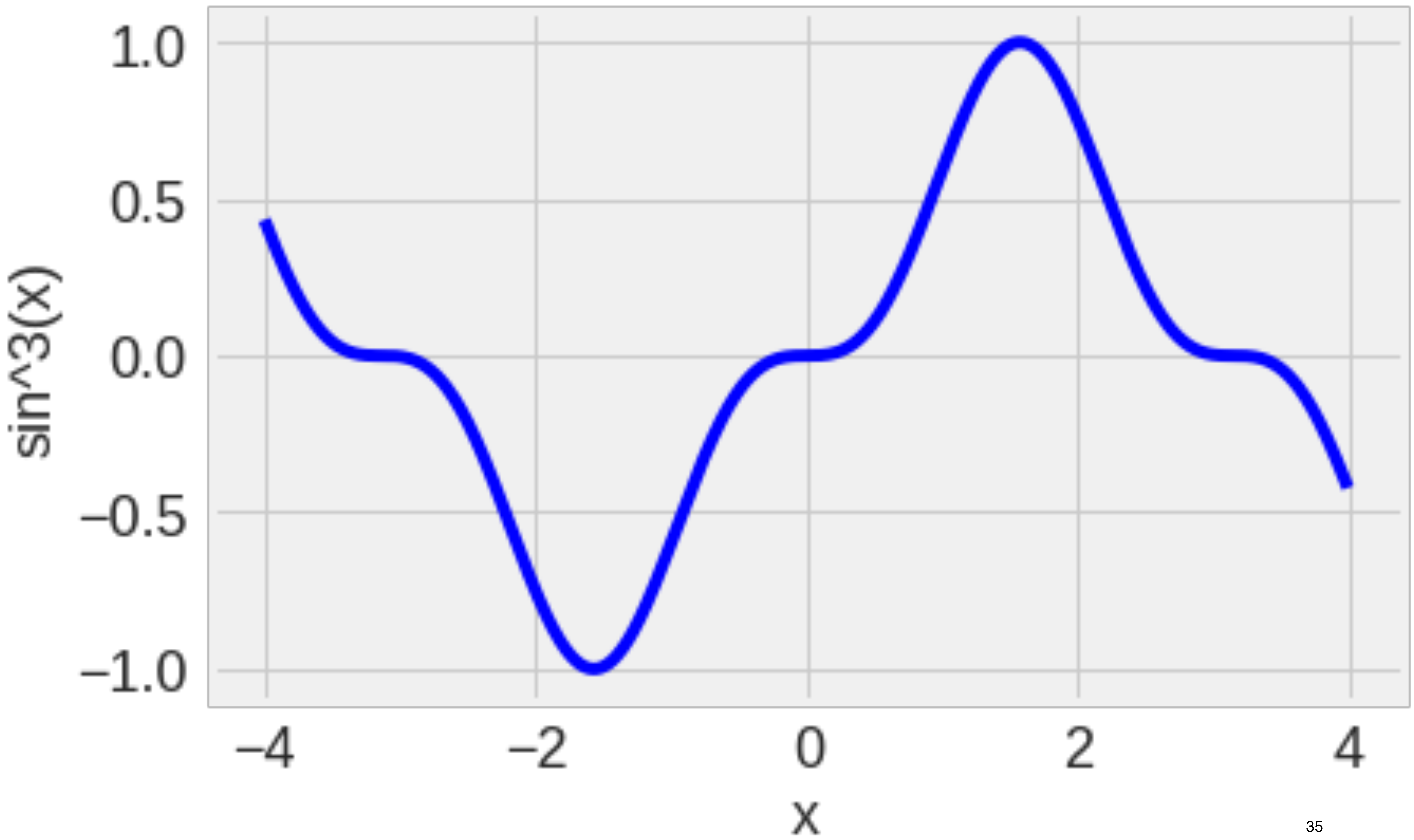
Try in colab!

```
import numpy as np
import matplotlib
from matplotlib import pyplot as plt
matplotlib.rc('axes', labelsize=18)
matplotlib.rc('xtick', labelsize = 18)
matplotlib.rc('ytick', labelsize = 18)
```

- Make plots with pyplot

```
x = np.arange(-4.0, 4.0, 0.01)
y = np.sin(x)**3
```

```
plt.clf() #clear figure
plt.plot(x,y, color='b')
plt.xlabel('x')
plt.ylabel('sin^3(x)')
plt.show()
```


1.0
0.5
0.0
−0.5
−1.0
sin^3(x)
−4
−2
0
2
4
x

Functions

Try in tutor!

```
def square(x):
  return x*x
```

```
square(4)
16
```

- Input(s) ("arguments")
- Returns output
- Functions can call other functions

Plotting π 1

- **Activity: edit your code**
  - Make everything but the first two lines and last two lines inside a function that takes one input, throws
  - Instead of setting throws = 10, throws in an input to the function
  - Return the pi estimate
  - Have your print statement just use your function

```
import math
import random

hits = 0
throws = 10

i = 0
while i < throws:
    x = random.random()
    y = random.random()

    rSquared = x**2 + y**2
    if rSquared < 1:
        hits = hits + 1
    i = i + 1
pi = 4.0 * float(hits) / float(throws)
print(pi)
```

Solution 1

- **Activity: edit your code**
  - Make everything but the last two lines inside a function that takes one input, n
  - Instead of setting throws = 10, set throws=n
  - Return the pi estimate

```python
import math
import random

def estimatePi(throws):
  hits = 0
  i = 0
  while i < throws:
      x = random.random()
      y = random.random()

      rSquared = x**2 + y**2
      if rSquared < 1:
          hits = hits + 1
      i = i + 1

  pi = 4.0 * float(hits) / float(throws)
  return pi

print(estimatePi(1e4))
```

Plotting π 2

- **Activity: edit your code**
  - Make a list of different numbers of throws: 10, 100, 1000, … up to 1e7
  - Make an empty list called piEstimates
  - Loop over the list you made, estimating π for different numbers of throws

```python
import math
import random

def estimatePi(throws):
  # (same definition of estimatePi function here)
  return pi

print(estimatePi(1e4))
```

Plotting π solution 2

- **Activity: edit your code**
  - Make a list of different numbers of throws: 10, 100, 1000, … up to 1e7
  - Make an empty list called piEstimates
  - Loop over the list you made, estimating π for different numbers of throws

```python
import math
import random

def estimatePi(throws):
  # (same definition of estimatePi function here)
  return pi

throwsList = [1e1, 1e2, 1e3, 1e4, 1e5, 1e6, 1e7]
piEstimatesList = []
for throws in throwsList:
    piEstimatesList.append(estimatePi(throws))
print(piEstimatesList)
```

Plotting π 3

- **Activity: edit your code**
  - Don't print piEstimatesList
  - Instead, make a scatter plot of throwsList vs. piEstimatesList
  - Use a log scale on the x axis: plt.xscale('log')

```
import math
import random

def estimatePi(throws):
  # (same definition of estimatePi function here)
  return pi

throwsList = [1e1, 1e2, 1e3, 1e4, 1e5, 1e6, 1e7]
piEstimatesList = []
for throws in throwsList:
    piEstimatesList.append(estimatePi(throws))
print(piEstimatesList)
```

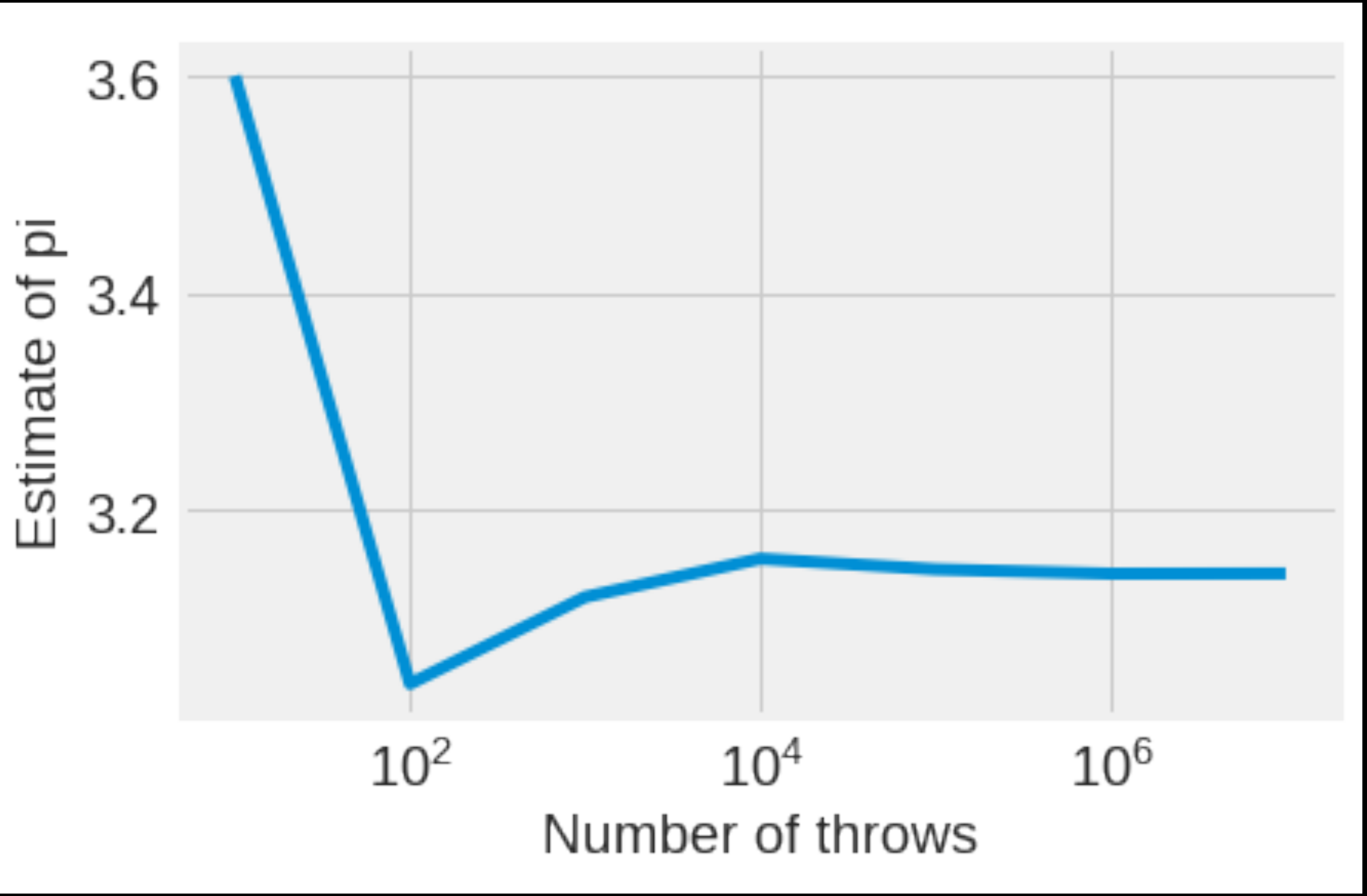
Solution 3

- **Activity: edit your code**
  - Don’t print piEstimatesList
  - Instead, make a scatter plot of throwsList vs. piEstimatesList
  - Use a log scale on the x axis: plt.xscale(‘log’)

```
import math
import random
import numpy as np
import matplotlib
from matplotlib import pyplot as plt
matplotlib.rc('axes', labelsize=18)
matplotlib.rc('xtick', labelsize = 18)
matplotlib.rc('ytick', labelsize = 18)

def estimatePi(throws):
  # (same definition of estimatePi function here)
  return pi

throwsList = [1e1, 1e2, 1e3, 1e4, 1e5, 1e6, 1e7]
piEstimatesList = []
for throws in throwsList:
    piEstimatesList.append(estimatePi(throws))

plt.clf()
plt.plot(throwsList, piEstimatesList)
plt.xlabel('Number of throws')
plt.xscale('log')
plt.ylabel('Estimate of pi')
plt.show()
```

Accuracy of the π dart board

- As throws goes up, answer gets closer to pi
- But it’s hard to see how close it is later on
- So instead, plot the difference between the estimate and the real answer



Plotting π 4

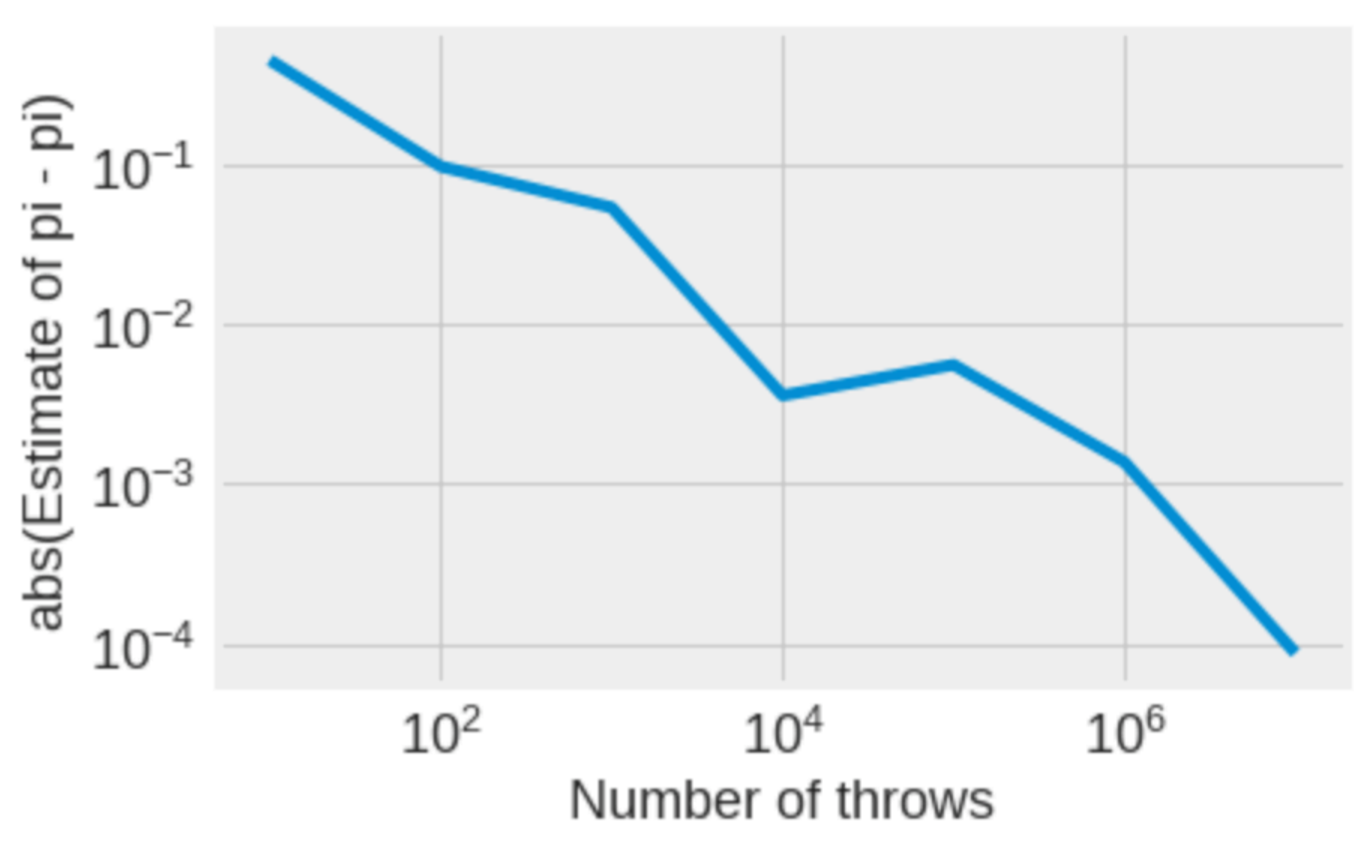
- **Challenge: edit your code**
  - import numpy as np
  - piEstimates = np.array(piEstimatesList)
  - Plot throwsList vs. abs(piEstimates - math.pi)
  - Put y axis on a log scale
  - Update y axis label to be abs(estimate of pi - pi)

```
# … code that computes pi

throwsList = [1e1, 1e2, 1e3, 1e4, 1e5, 1e6, 1e7]
piEstimatesList = []
for throws in throwsList:
    piEstimatesList.append(estimatePi(throws))

plt.clf()
plt.plot(throwsList, piEstimatesList)
plt.xlabel('Number of throws’)
plt.xscale('log')
plt.ylabel('Estimate of pi')
plt.show()
```

Solution 4

- **Challenge: edit your code**
  - import numpy as np
  - piEstimates = np.array(piEstimatesList)
  - Plot throwsList vs. abs(piEstimates - math.pi)
  - Put y axis on a log scale
  - Update y axis label to be abs(estimate of pi - pi)

```
# … code that computes pi

throwsList = [1e1, 1e2, 1e3, 1e4, 1e5, 1e6, 1e7]
piEstimatesList = []
for throws in throwsList:
    piEstimatesList.append(estimatePi(throws))

plt.clf()
piEstimates = np.array(piEstimatesList)
plt.plot(throwsList, abs(piEstimates - math.pi))
plt.xlabel('Number of throws')
plt.xscale('log')
plt.yscale('log')
plt.ylabel('abs(Estimate of pi – pi)')
plt.show()
```



abs(Estimate of pi - pi)
10−1
10−2
10−3
10−4
10²
10⁴
10⁶
Number of throws

Concepts in numerical programming

- Resolution
- Accuracy
- Precision

Resolution

Low resolution

Entire image: 227KB

Large galaxies
1 billion light years away

Small galaxies up to
13 billion light years away

Image courtesy NASA

Resolution

High resolution

Entire image: 110MB

Large galaxies
1 billion light years away

Small galaxies up to
13 billion light years away

Image courtesy NASA

Resolution

- **Low resolution**
  - Smaller data
  - Faster computation
  - Less precise
- **High resolution**
  - Bigger data
  - Slower computation
  - More precise

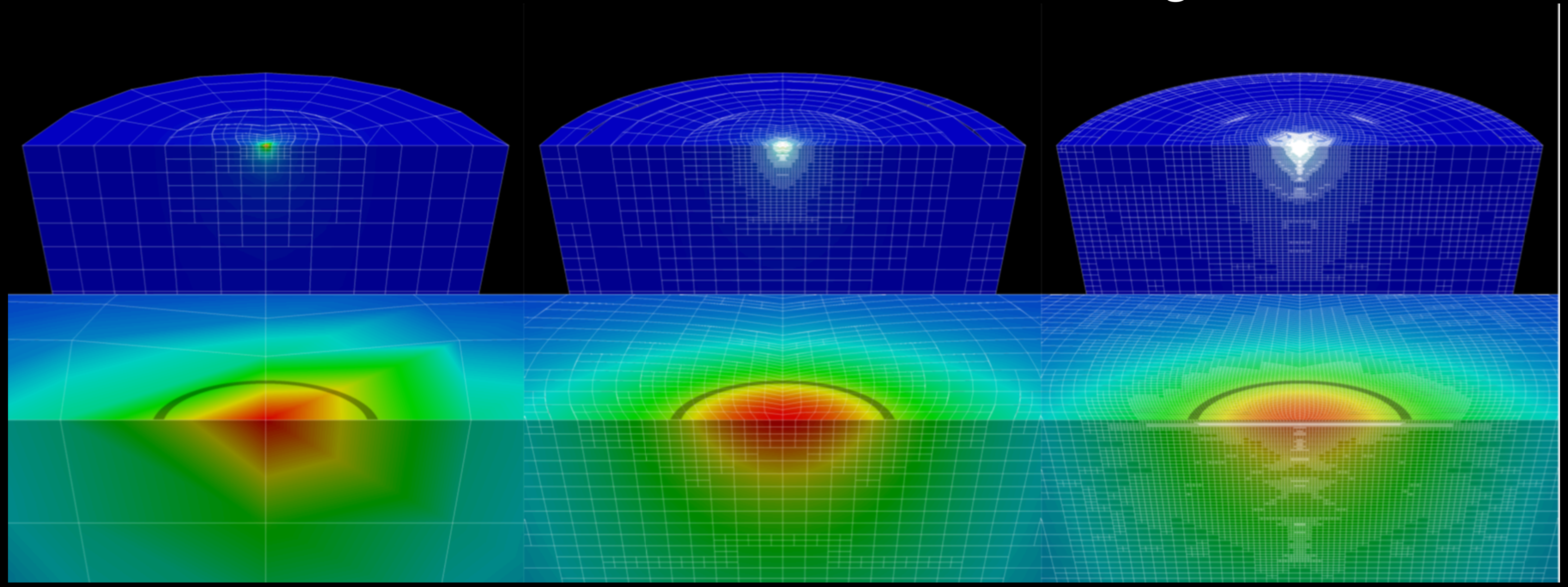
Precision & accuracy

- **Precision**
  - How much result changes when you add more resolution
  - "How many digits"
- **Accuracy**
  - How close result is to the correct result

Example: thermal noise

- Thermal noise of a mirror in LIGO depends on how much potential energy it gets when you push on the face

Low resolution *Medium resolution* *High resolution*

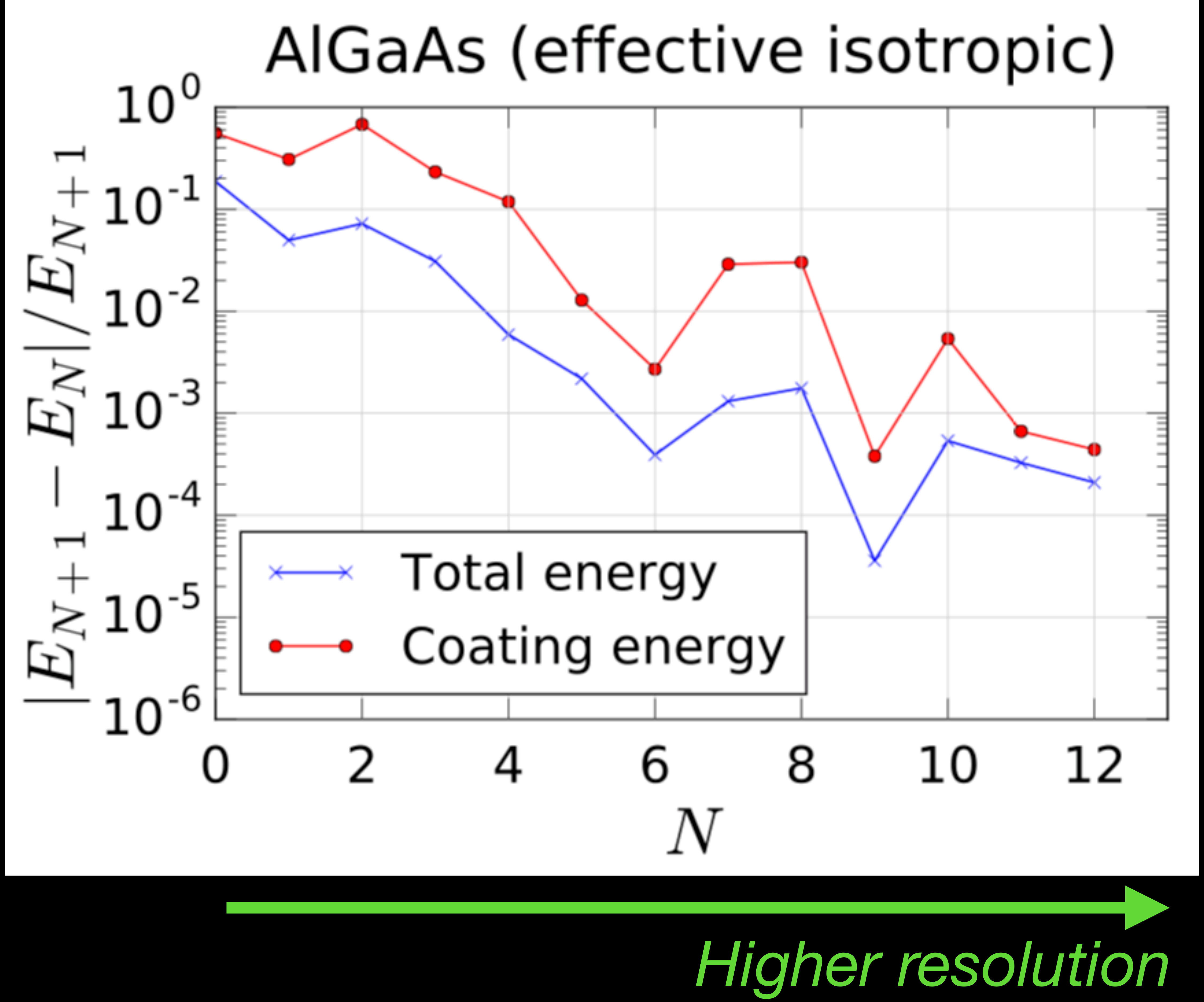


Color = how much mirror deforms

Lovelace, Demos, Khan (2018)

Example: thermal noise

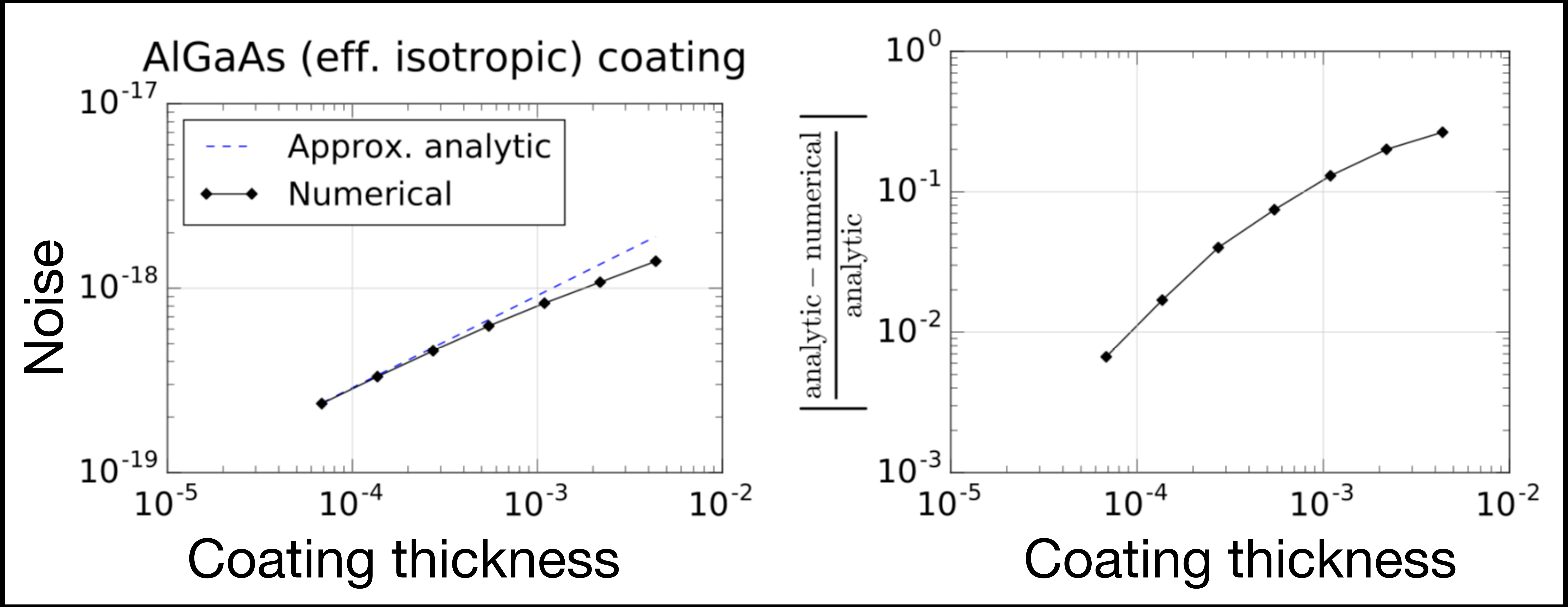
- Potential energy E in deformed mirror
  - Precision of energy as resolution increases
    - Label resolution by integer N



Lovelace, Demos, Khan (2018)

Example: thermal noise

- Thermal noise of thin coating
  - Accuracy: compare code to known "analytic" solution



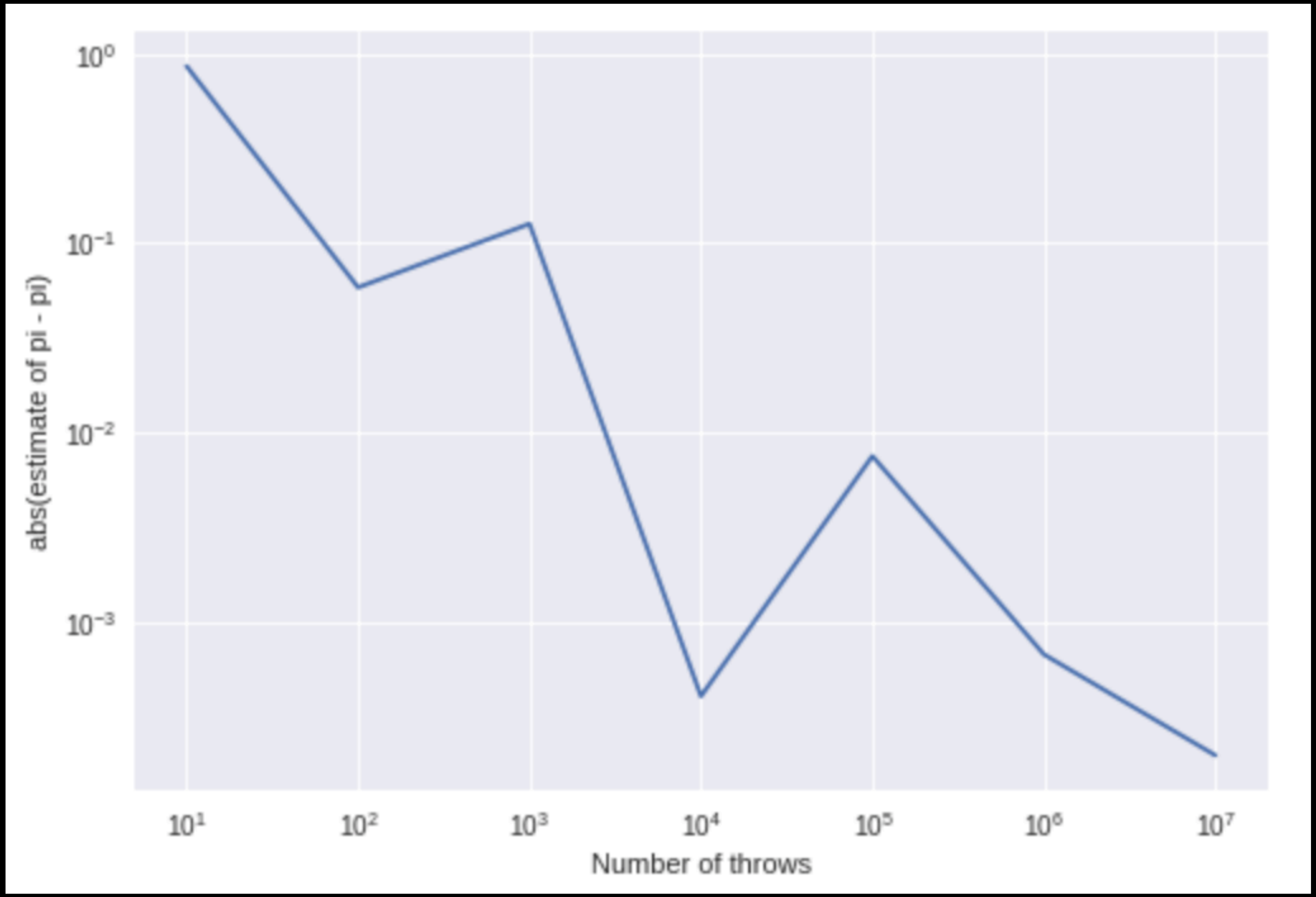
Lovelace, Demos, Khan (2018)

Unix terminal on the web

- Go to https://mybinder.org/v2/gh/geoffrey4444/NRDataExample/master
- Choose "New Terminal"

Clicker question #2.7a

- Your graph plots abs(estimate of pi - pi) vs number of throws. The **horizontal axis** shows



A Precision

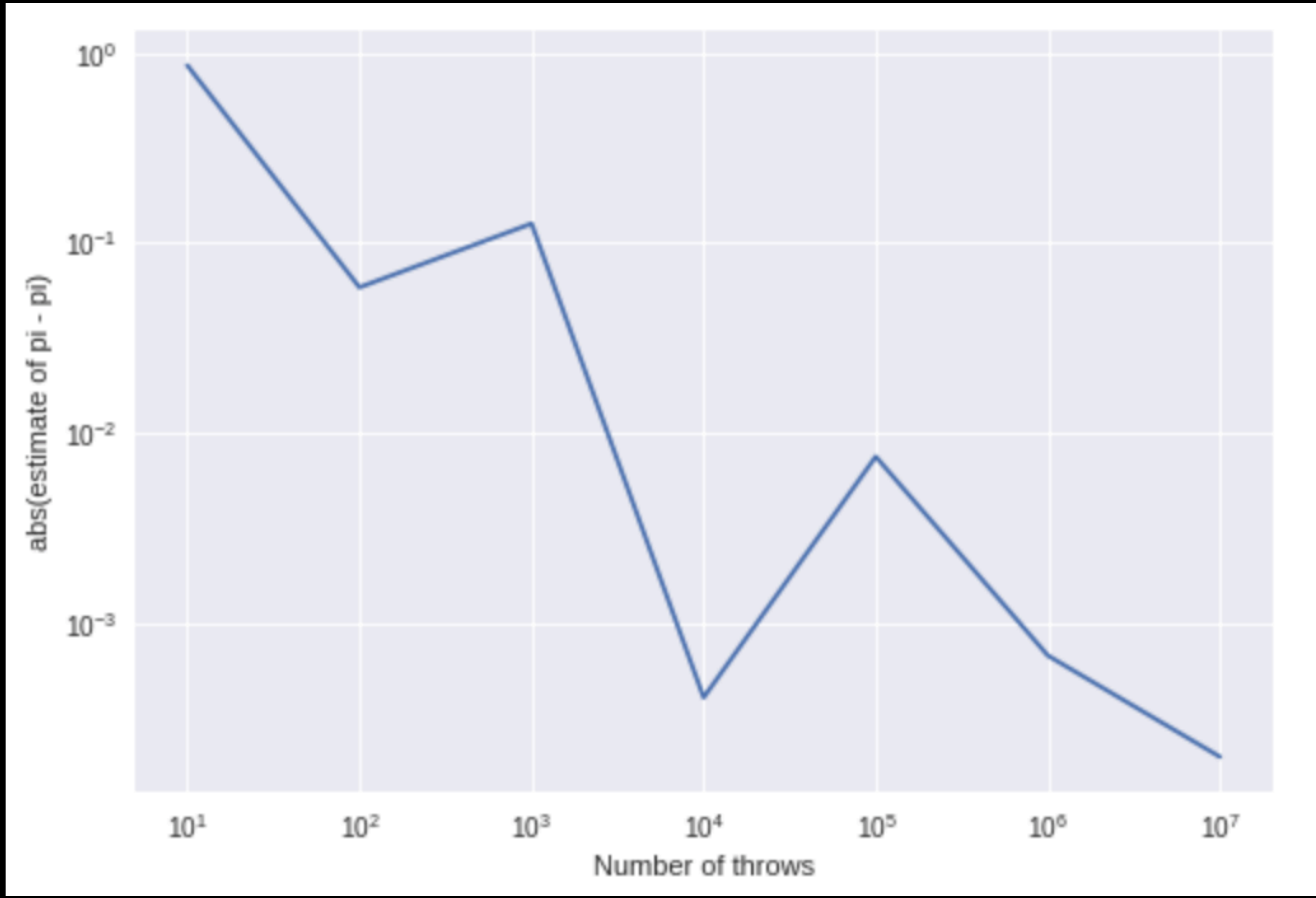
B Accuracy

C Resolution

D None of ABC

Clicker question #2.7b

- Your graph plots abs(estimate of pi - pi) vs number of throws. The **vertical axis** shows



A Precision

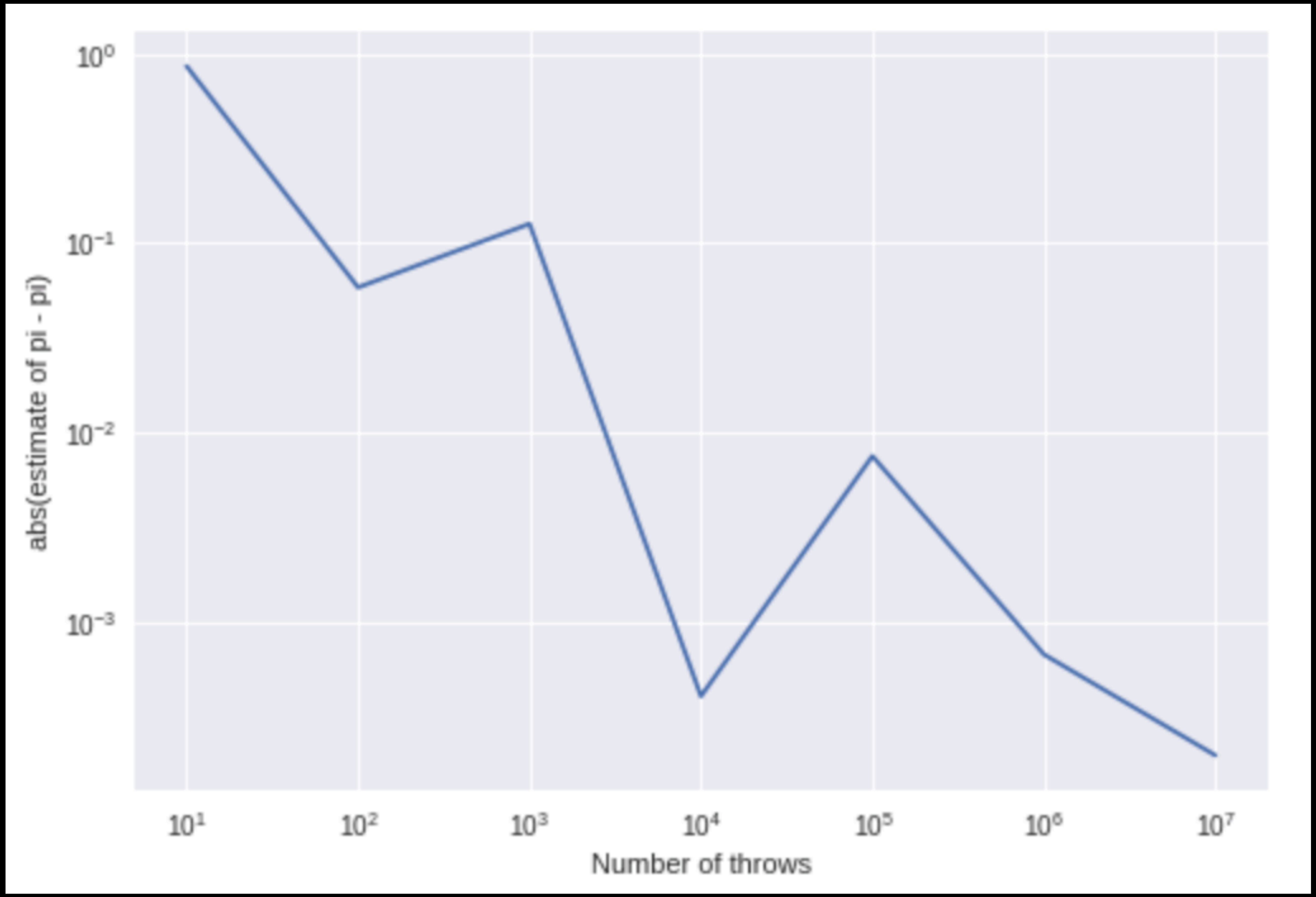
B Accuracy

C Resolution

D None of ABC

Clicker question #2.7c

- As the number of throws increases, the **resolution**



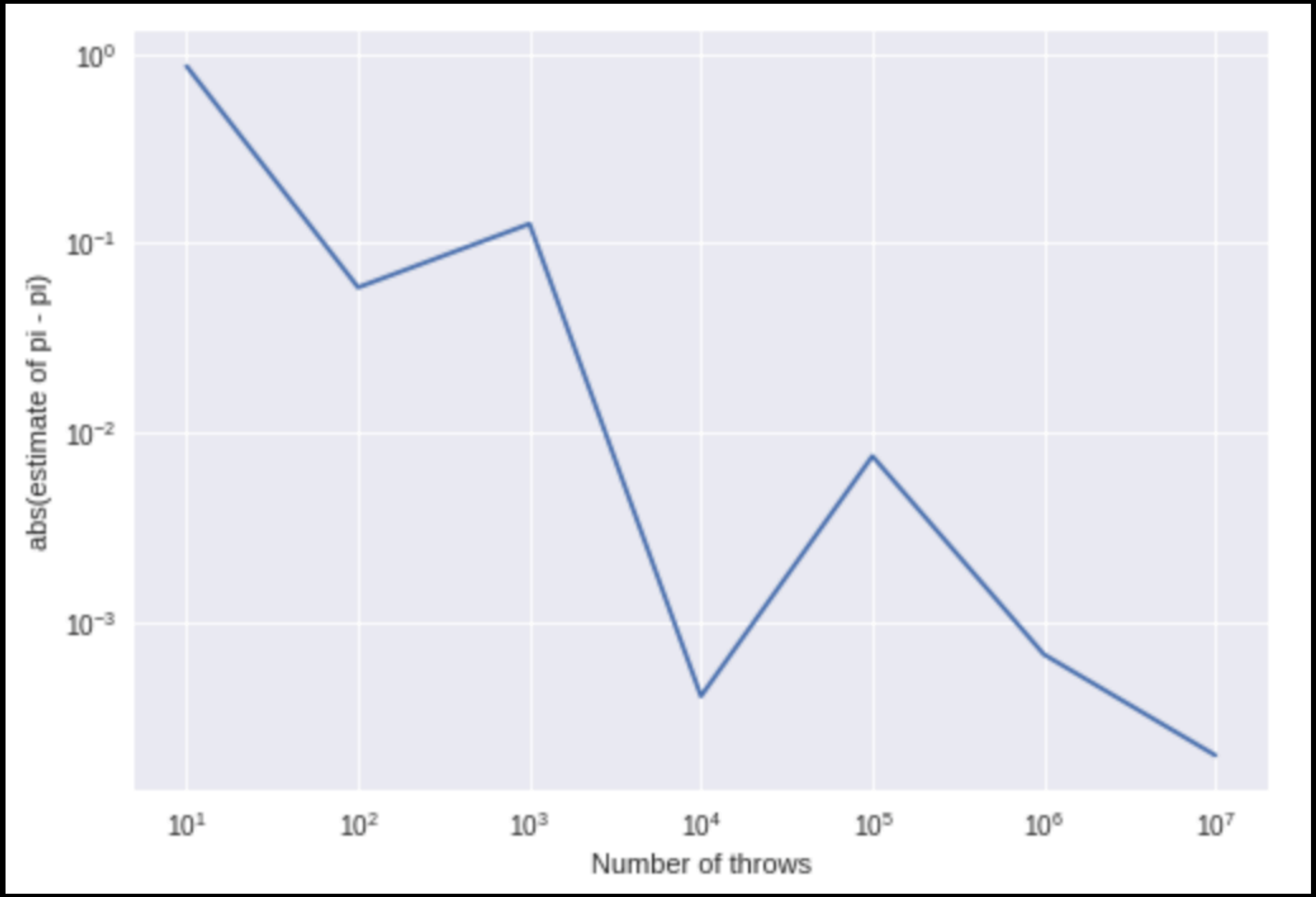
A Increases

B Decreases

C Stays the same

Clicker question #2.7d

- As the number of throws increases, the **accuracy**



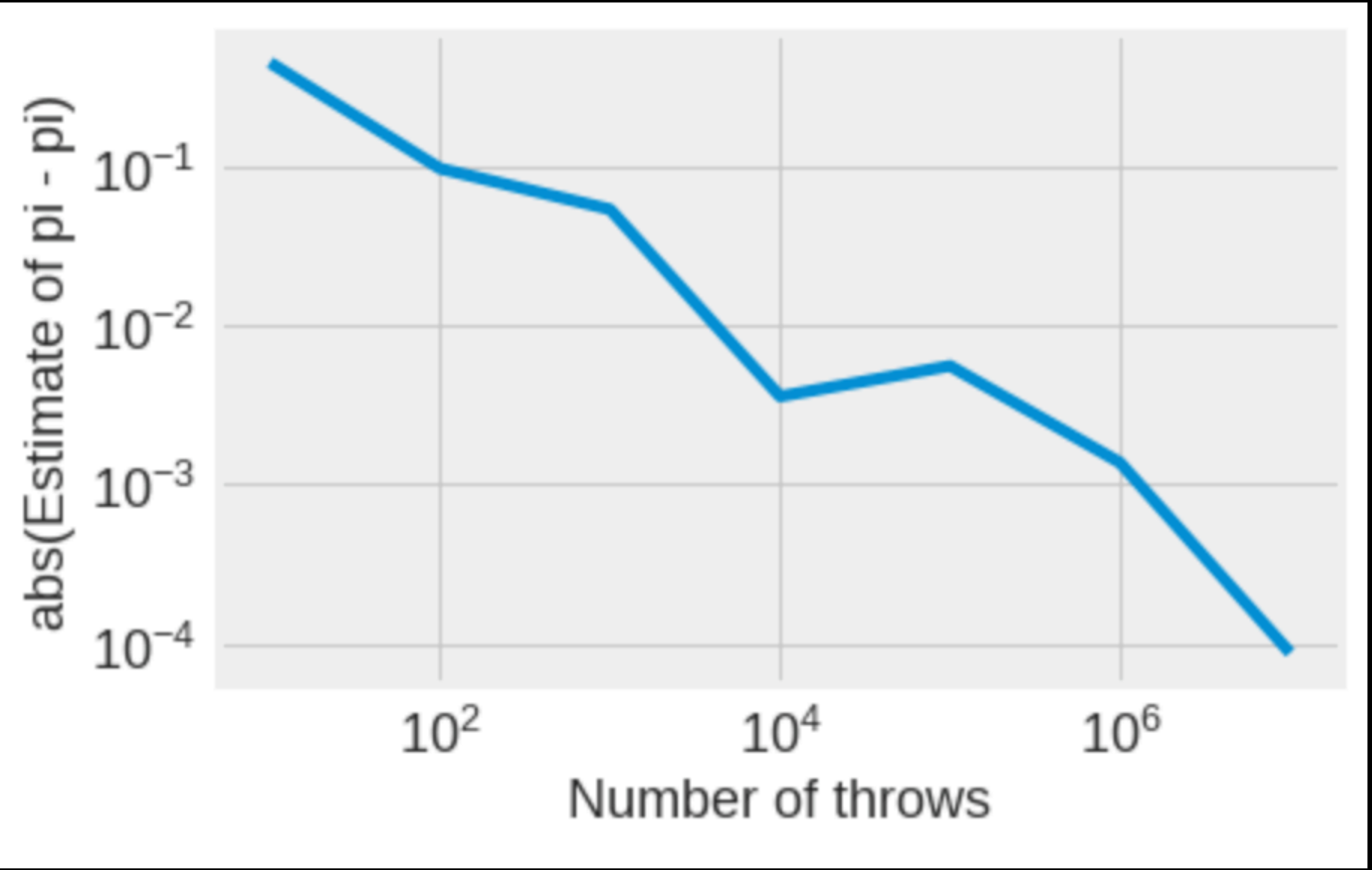
A Increases

B Decreases

C Stays the same

Example: π dart board

- Need roughly 100x more darts to get 10x more accuracy
  - That is, 100x darts gives you 10x more accuracy
  - This gets slow fast!
  - Can we do better?



Connect to ocean (use our laptops)

- Open "PuTTY 64-bit" on the desktop
- Under "saved sessions" select "ocean"
- Click "Open"
- Username: workshopStudents2019
- Passphrase
  - Some computers: workshop2019!!
- `squeue`
- `srun -p orca-1 --nodes=1 --tasks-per-node=20 --pty /bin/bash`

UNIX command line crash course activity

- **Commands to know**
  - ls, pwd, cd, mkdir
  - ./, ../, paths
  - cp, mv, rm, rmdir
  - cat, less
  - nano
  - whoami, date, …

- **Play along with me…**
  - mkdir YOURNAME and cd into it
  - Navigate file system: ls, pwd, cd, ./ and ../
  - Use nano to write a text file
  - Copy, rename, remove a file
  - Cat, less, more, head, tail
  - > to redirect output
  - Grab bag: whoami, date, grep, sed, zip…

Unix commands to know

- **Commands to know**
  - ls, pwd, cd, mkdir
  - ./, ../, paths
  - cp, mv, rm, rmdir
  - cat, less
  - nano
  - whoami, date, …

- **Play along with me…**
  - mkdir YOURNAME and cd into it
  - Navigate file system: ls, pwd, cd, ./ and ../
  - Use nano to write a text file
  - Copy, rename, remove a file
  - Cat, less, more, head, tail
  - > to redirect output
  - Grab bag: whoami, date, grep, sed, man, …

Clicker question #1.8

- I want to list the files in the directory I'm in. Which command would I use?

A	ls
B	cd
C	pwd
D	nano

Clicker question #1.9

- Which command edits the file “Hello.txt” in the directory I am currently in?

A nano ./Hello.txt

B cat ./Hello.txt

C nano ../Hello.txt

D cat ../Hello.txt

Clicker question #1.9

- Which command makes a new directory called "TestFolder"?

A ls TestFolder

B cd TestFolder

C mkdir TestFolder

D cp TestFolder

Clicker question #1.9

- Which command removes everything in the current directory, which is not empty?

A rmdir ./

B rm -r ./*

C rm -r . /*

D More than one of these will work

Unix activity

- **Commands to know**
  - ls, pwd, cd, mkdir
  - ./, ../, paths
  - cp, mv, rm, rmdir
  - cat, less
  - nano
  - whoami, date, …

- Use nano to write a bash script (each line is a command like you would enter on the command line)
- The script should…
  - Print the current date and time
  - Print the current directory
  - Copy /proc/cpuinfo into the current directory
  - Get the first line of the copied file, and save it to a file called FirstLineOfProc.txt
  - Bonus: Use grep to only show the line with "cpu cores"
  - Bonus: use sed to remove all but the core number
  - Bonus: instead of copying the /proc/cpuinfo file, copy whatever file users specify as an argument (google bash arguments)

Parallel computing

- Supercomputers have lots of cores
- But each core is not much faster than a PC
- To take full advantage, you have to write code that can run on more than one core at the same time
  - That is, code that runs in parallel

Image courtesy Blue Waters

Connect to ocean

- Open "PuTTY 64-bit" on the desktop
- Under "saved sessions" select "ocean"
- Click "Open"
- Username: workshopStudents2019
- Passphrase
  - Some computers: workshop2019!!
- `squeue`
- `srun -p orca-1 --nodes=1 --tasks-per-node=20 --pty /bin/bash`

Parallel computing 1

- Log into ocean
- Do this

```
#Replace GeoffreyLovelace with YourName
cd student_folders
mkdir YOUR_NAME #replace with your lastName_firstName
cd YOUR_NAME

mkdir PiDart
cd PiDart
source /opt/ohpc/pub/apps/anaconda2/bin/activate root
```

Parallel computing 2

- `nano Hello.py`

```
print(“Hello")
```

- `mpirun -np 8 python Hello.py`
  - **What happens? What happens if you change 8 to another number less than 8?**

What happened?

- mpirun ran many copies of “Hello.py”
- Each copy printed “Hello”
  - But the processors are not working together yet, or even doing anything different
- Next: make different processors do different things

Parallel computing 3

- `cp Hello.py MpiHello.py`
- `nano MpiHello.py`

```python
from mpi4py import MPI
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()
print("Hello from processor "+str(rank)+" out of "+str(size))
```

- `mpirun -np 4 python MpiHello.py`
- `mpirun -np 8 python MpiHello.py`