XSEDE Code of Conduct

- XSEDE has an external code of conduct for XSEDE sponsored events which represents XSEDE's commitment to providing an inclusive and harassment-free environment in all interactions regardless of gender, sexual orientation, disability, physical appearance, race, or religion. The code of conduct extends to all XSEDE-sponsored events, services, and interactions.
- **Code of Conduct:** https://www.xsede.org/codeofconduct
- Contact:
  - Event organizer: Rosalia Gomez (rosie@tacc.utexas.edu)
  - XSEDE ombudspersons:
    - Linda Akli, Southeastern Universities Research Association, (akli@sura.org)
    - Lizanne Destefano, Georgia Tech (lizanne.destefano@ceismc.gatech.edu)
    - Ken Hackworth, Pittsburgh Supercomputing Center, (hackworth@psc.edu)

XSEDE

Extreme Science and Engineering Discovery Environment

Python Programming in Spyder

Presented by:

- Kate Cahill, Ohio Supercomputer Center (kcahill@osc.edu)
- Je'aime Powell, Texas Advanced Computing Center (jpowell@tacc.utexas.edu)

Goals for this session

Gain practice writing Python functions

Define Monte Carlo method and how to mathematically estimate π

Iteratively build a Python program to solve a problem

Monte Carlo sampling method

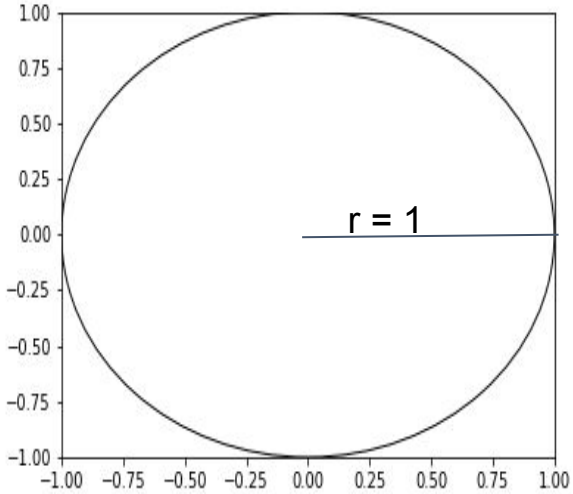
A method of estimating the value of an unknown quantity using a random sample of a population

Used in chemistry, astrophysics, engineering research and many other disciplines

Front of the Grand Casino I Monte Carlo, Monaco

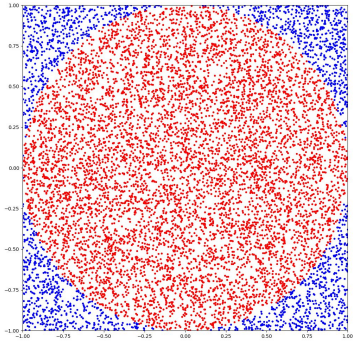
How can we estimate pi mathematically?

- We have a circle inscribed in a square. The center of the circle is located at 0,0.
- The circle has a radius of 1 unit. The area of the circle is $\pi * r^2 = \pi * 1 = \pi$
- The square is 2 units long on each side. The area of the square is $l^2 = 2^2 = 4$
- The probability of random points being inside the circle is equal to the area of the circle / the area of square:
  a. P(point inside circle) = π / 4
- Solve for π: (Points inside circle / total points) * 4 = π



Why should we? Because we already know the answer! That makes it easy to check if we got it right

Pseudocode



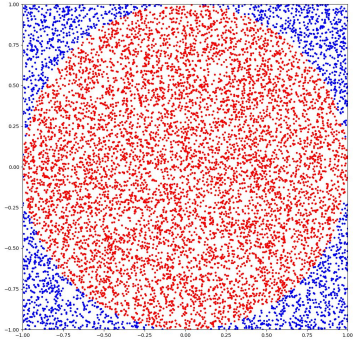
Think through the problem!

Write down the steps that will solve the problem

Use this as an outline of your code

1. Throw darts
2. Count darts that fall inside circle
3. Calculate Pi
4. Repeat
5. Get estimate as well as mean, standard deviation

Throwing darts



How can we randomly sample this space?

The Numpy library has a random number generator

Automatically generates numbers

Usage:

Import numpy as np

np.random.random((100, 2))

- This will generate 100 sets of 2 numbers

Change range to fill circle

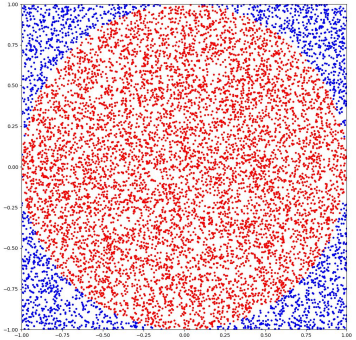
Np.random.random generates numbers inside the set (0,1]

We need the set (-1,1]

(0,1] * 2 = (0, 2]

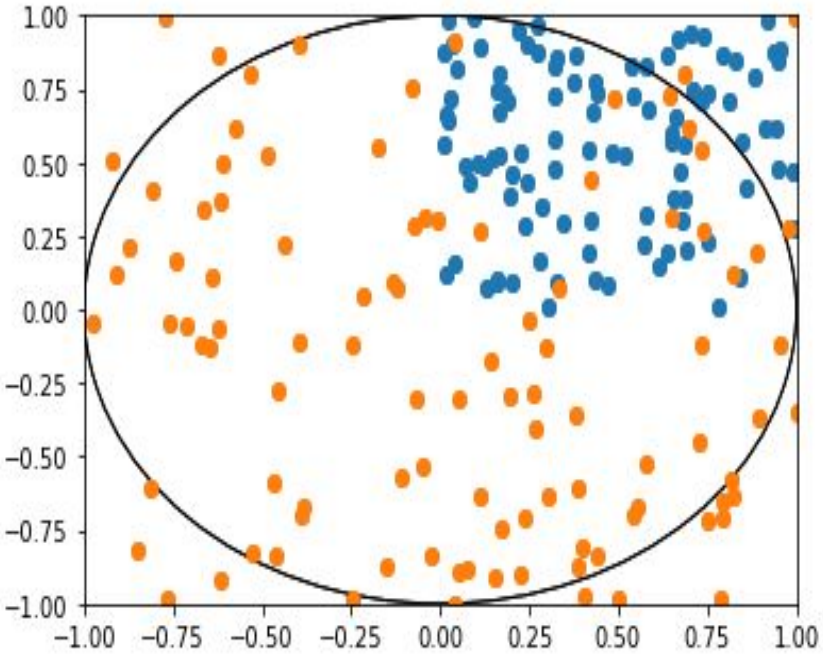
(0, 2] - 1 = (-1, 1]

2 * np.random.random((100, 2)) - 1



```
x = darts[:,0]
y = darts[:,1]
```

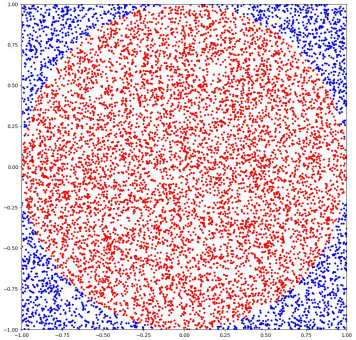
```
place_darts = plt.scatter(x,y)
ax = plt.gca()
ax.set_xlim((-1, 1))
ax.set_ylim((-1, 1))
circle2 = plt.Circle((0, 0), 1,
fill=False)
ax.add_artist(circle2)
ax.add_artist(place_darts)
```



The dots in blue only cover the range (0,1)

The dots in orange are after we expanded the range to (-1,1)

Are they in or are they out?



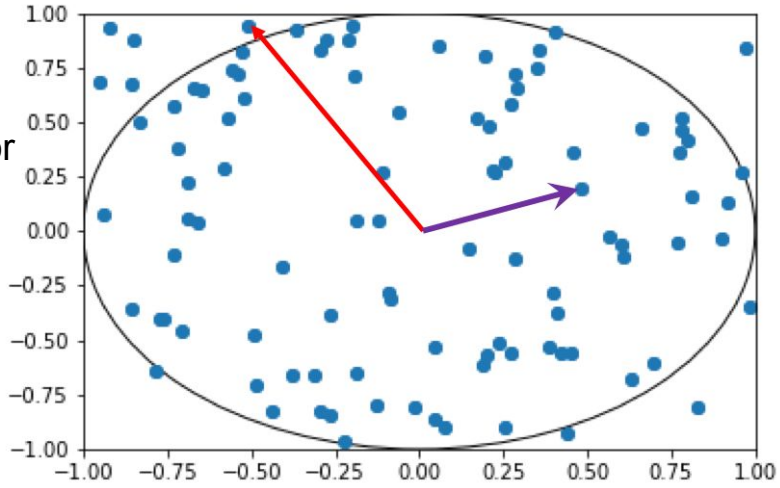
We have to determine which darts are inside the circle

We can do this by measuring the length of the vector from 0,0

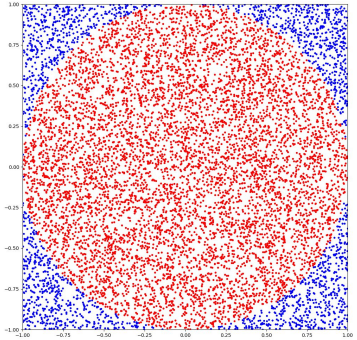
Two possibilities:

Use pythagorean theorem to calculate length of the vector
m.sqrt(x2 + y2) < 1.0

Use Numpy command linalg.norm
np.linalg.norm(p,axis=-1)<=1.0



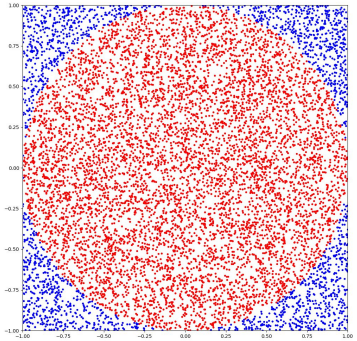
Perform the calculation



To estimate pi, we need to calculate the ratio of darts inside the circle to all darts thrown

4 * (Ndarts_in_circle / Total_darts) = pi estimate

How many points do we need?



What's a good sample size?

Is more always better?

What are the limits of processing capacity/time?

How can we take advantage of parallel processing?

More points, more runs!

Break out session

Please use your video

We will review the exercises in the handout

http://hpcuniversity.org/trainingMaterials/249/