Introduction to Stochastic Simulation

Python 5

General Plan

Assignments. Any issue with the last assignment? How long did it take you?

Project. See Canvas page.

— By pairs: same project programmed in Mathematica and in Python

— Choice from a list, or project of your own choice (upon approval)

— Work on the project during the 6 sessions of weeks 6 and week 7

— Deadline at the end of week 7

Plan.

0. Feedback on last week's exercises
1. Random Number Generators
2. Sampling Random variables
3. Example Brownian motion
4. Introduction to Monte Carlo

Part 0
Last week's exercises

Do you have any question?

Is there anything you are not sure you have understood?

Part 1

Random Number Generators

Random Number Generators

Question: How are random numbers generated in a computer?

Random Number Generators

Question: How are random numbers generated in a computer?

Hardware random number generators, or **True random number generators:**

Devices that generate random numbers from measurements of a physical process expected to be random (e.g., thermal noise, atmospheric noise, quantum phenomena)

Pseudo-random number generators:

algorithms that can generate long sequences of numbers that are random in appearance, but that are completely determined by the initial value, the **seed.**

Sequence has a finite length = period of the generator —> can sometimes be an issue

Pseudo-random Number Generators in NumPy

Using np.random :

Uses by default a **Mersenne-Twister generator**:

— one of the most tested random number generator
— has a very long period of 2^{19937}-1

>>> Ex1-2

```
np.random.random()
```

```
0.9926001562825242
```

```
np.random.random((3,2))
```

```
array([[0.73093464, 0.06498689],
       [0.83402947, 0.73218955],
       [0.08812384, 0.22933893]])
```

Pseudo-random Number Generators in NumPy

Using np.random :

Uses by default a **Mersenne-Twister generator**:

— one of the most tested random number generator
— has a very long period of 2^{19937}-1

```
np.random.random()
```

```
0.9926001562825242
```

>>> Ex1-2

```
np.random.random((3,2))
```

```
array([[0.73093464, 0.06498689],
       [0.83402947, 0.73218955],
       [0.08812384, 0.22933893]])
```

```
np.random.seed(12)
np.random.random((3,2))
```

Setting the seed

```
array([[0.15416284, 0.7400497 ],
       [0.26331502, 0.53373939],
       [0.01457496, 0.91874701]])
```

>>> Q1

Pseudo-random Number Generators in NumPy

"Generator" object : new recommended syntax

```
### Without specifying the seed:
rng=np.random.default_rng()
print(rng)
rng.random((3,3))
```

Define the generator

```
Generator(PCG64)
```

By default: PCG64

```
array([[0.7493631 , 0.21492898, 0.97161296],
       [0.96073753, 0.55448631, 0.82254493],
       [0.2566791 , 0.51780946, 0.6871649 ]])
```

>>> Ex 3

```
### With the seed:
rng=np.random.default_rng(seed=42)
rng.random((3,3))
```

Setting the seed

```
array([[0.77395605, 0.43887844, 0.85859792],
       [0.69736803, 0.09417735, 0.97562235],
       [0.7611397 , 0.78606431, 0.12811363]])
```

Pseudo-random Number Generators in NumPy

"Generator" object : new recommended syntax

```
### Without specifying the seed:
rng=np.random.default_rng()
print(rng)
rng.random((3,3))
```

Define the generator

```
Generator(PCG64)
```

By default: PCG64

```
array([[0.7493631 , 0.21492898, 0.97161296],
       [0.96073753, 0.55448631, 0.82254493],
       [0.2566791 , 0.51780946, 0.6871649 ]])
```

>>> Ex 3

```
### With the seed:
rng=np.random.default_rng(seed=42)
rng.random((3,3))
```

Setting the seed

```
array([[0.77395605, 0.43887844, 0.85859792],
       [0.69736803, 0.09417735, 0.97562235],
       [0.7611397 , 0.78606431, 0.12811363]])
```

```
### Using Mersenne-Twister generator, MT19937:
from numpy.random import Generator, MT19937

rng=np.random.Generator(MT19937())
print(rng)
print(rng.random())
```

```
Generator(MT19937)
0.34337016943942145
```

>>> Q2

Pseudo-random Number Generators in NumPy

"Generator" object : new recommended syntax

```
### Without specifying the seed:
rng=np.random.default_rng()
print(rng)
rng.random((3,3))
```

Define the generator

```
Generator(PCG64)
```

By default: PCG64

```
array([[0.7493631 , 0.21492898, 0.97161296],
       [0.96073753, 0.55448631, 0.82254493],
       [0.2566791 , 0.51780946, 0.6871649 ]])
```

>>> Ex 3

```
### With the seed:
rng=np.random.default_rng(seed=42)
rng.random((3,3))
```

Setting the seed

```
array([[0.77395605, 0.43887844, 0.85859792],
       [0.69736803, 0.09417735, 0.97562235],
       [0.7611397 , 0.78606431, 0.12811363]])
```

```
### Using Mersenne-Twister generator, MT19937:
from numpy.random import Generator, MT19937

rng=np.random.Generator(MT19937())
print(rng)
print(rng.random())
```

>>> Q2

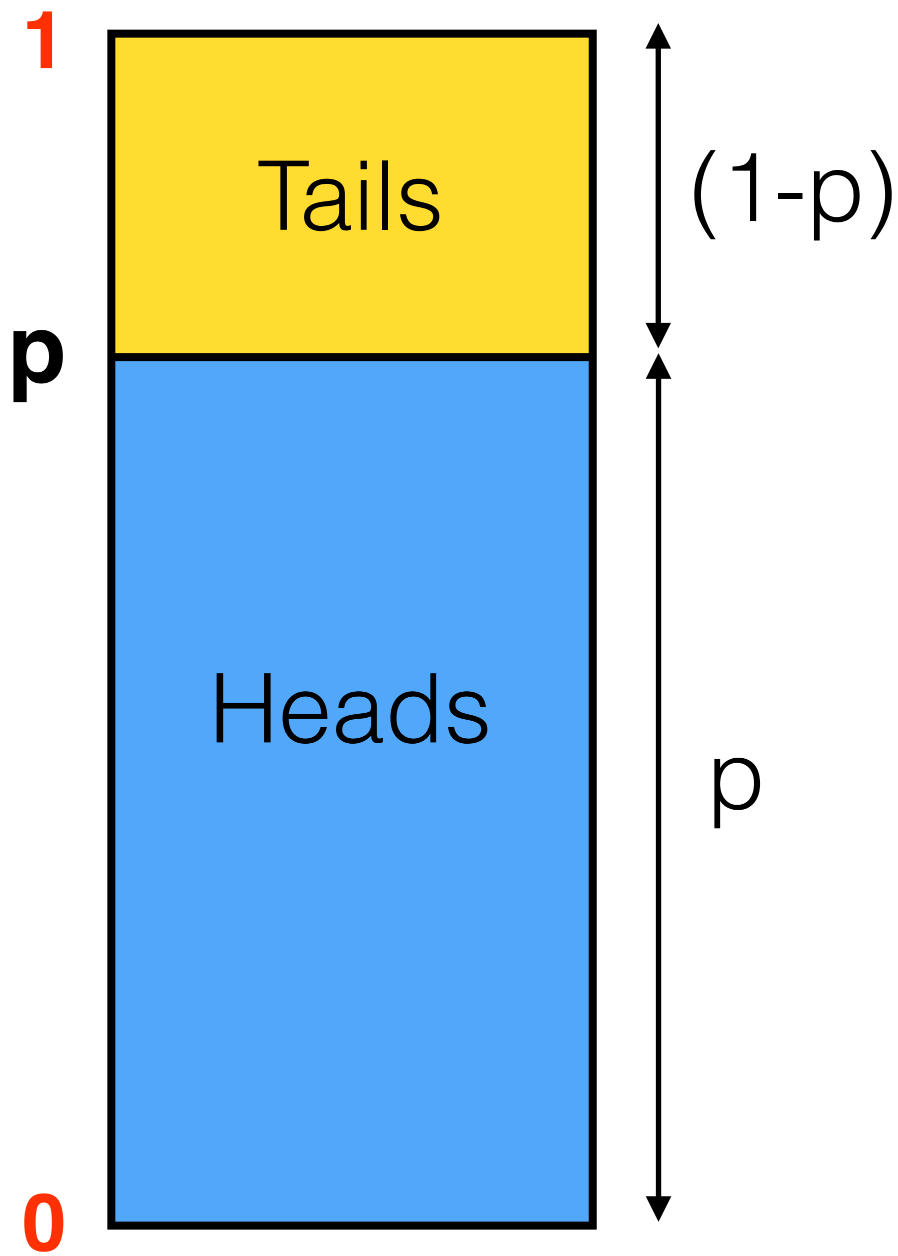
>>> Q3-4

```
Generator(MT19937)
0.34337016943942145
```

Part 2

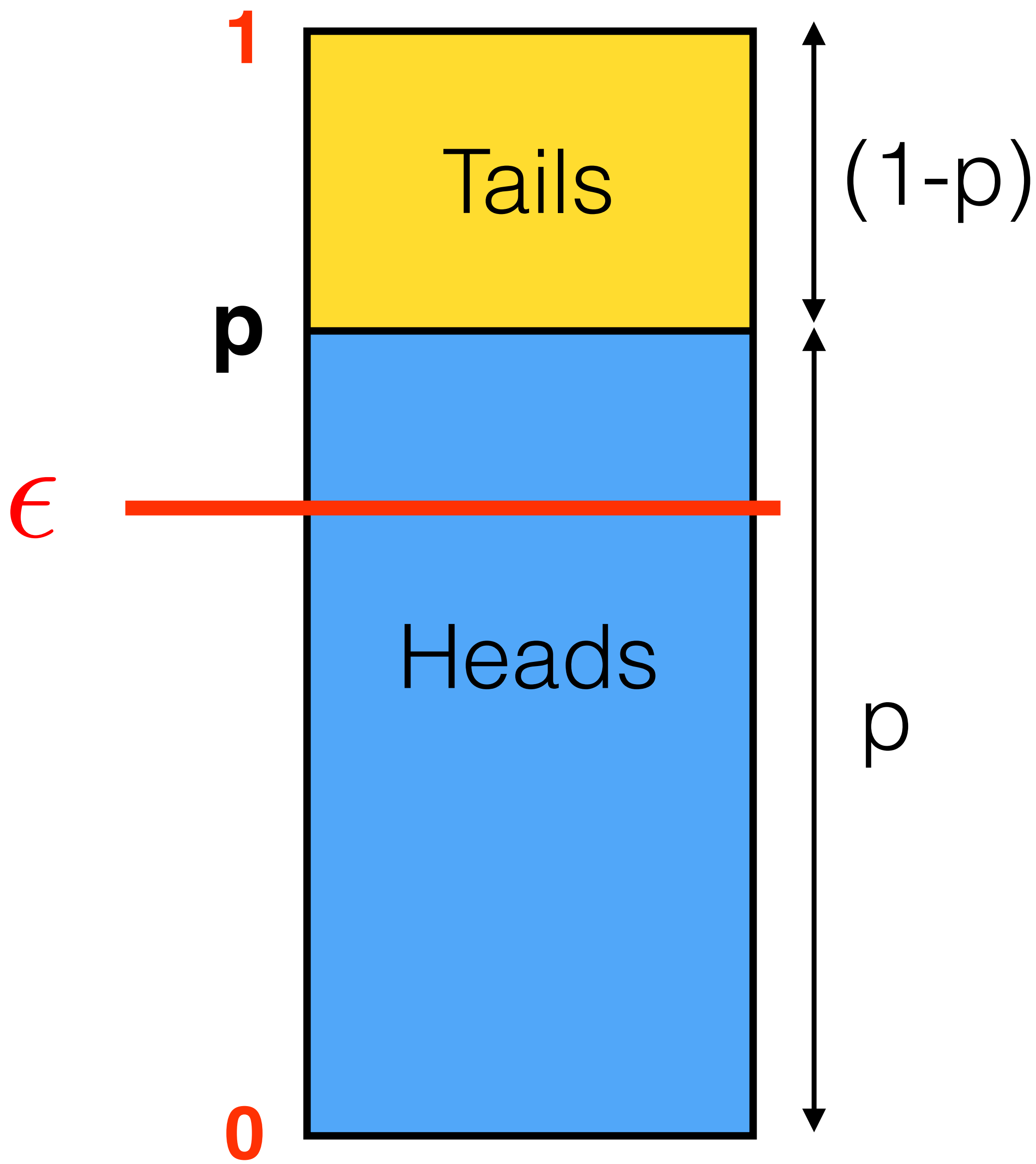
Sampling random variables

Sampling a Binary Discrete Random Variable



Ex. Unfair coin:

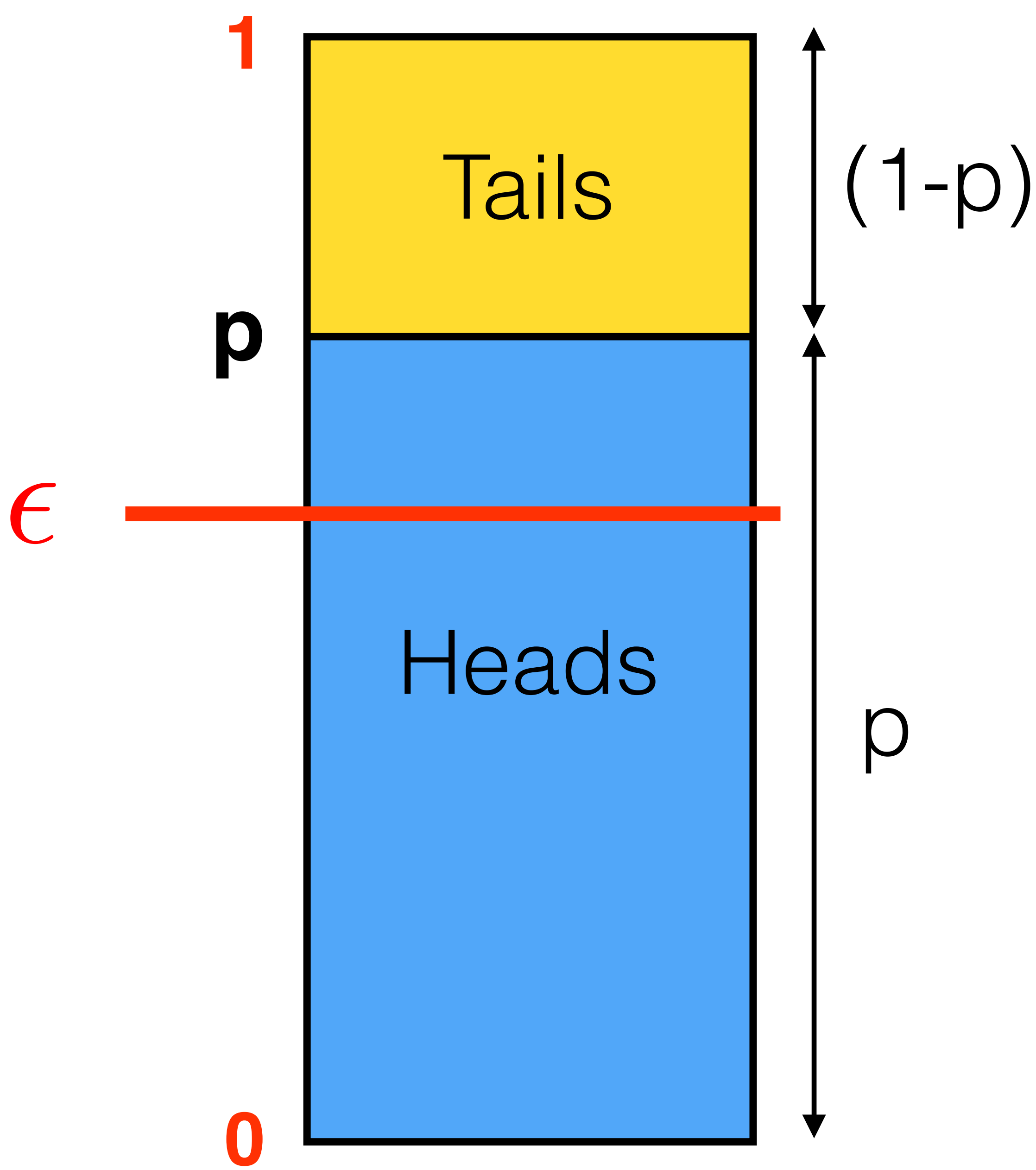
Sampling a Binary Discrete Random Variable



Ex. Unfair coin:

$\epsilon \in [0, 1)$ uniform

Sampling a Binary Discrete Random Variable



Ex. Unfair coin:

$\epsilon \in [0, 1)$ uniform

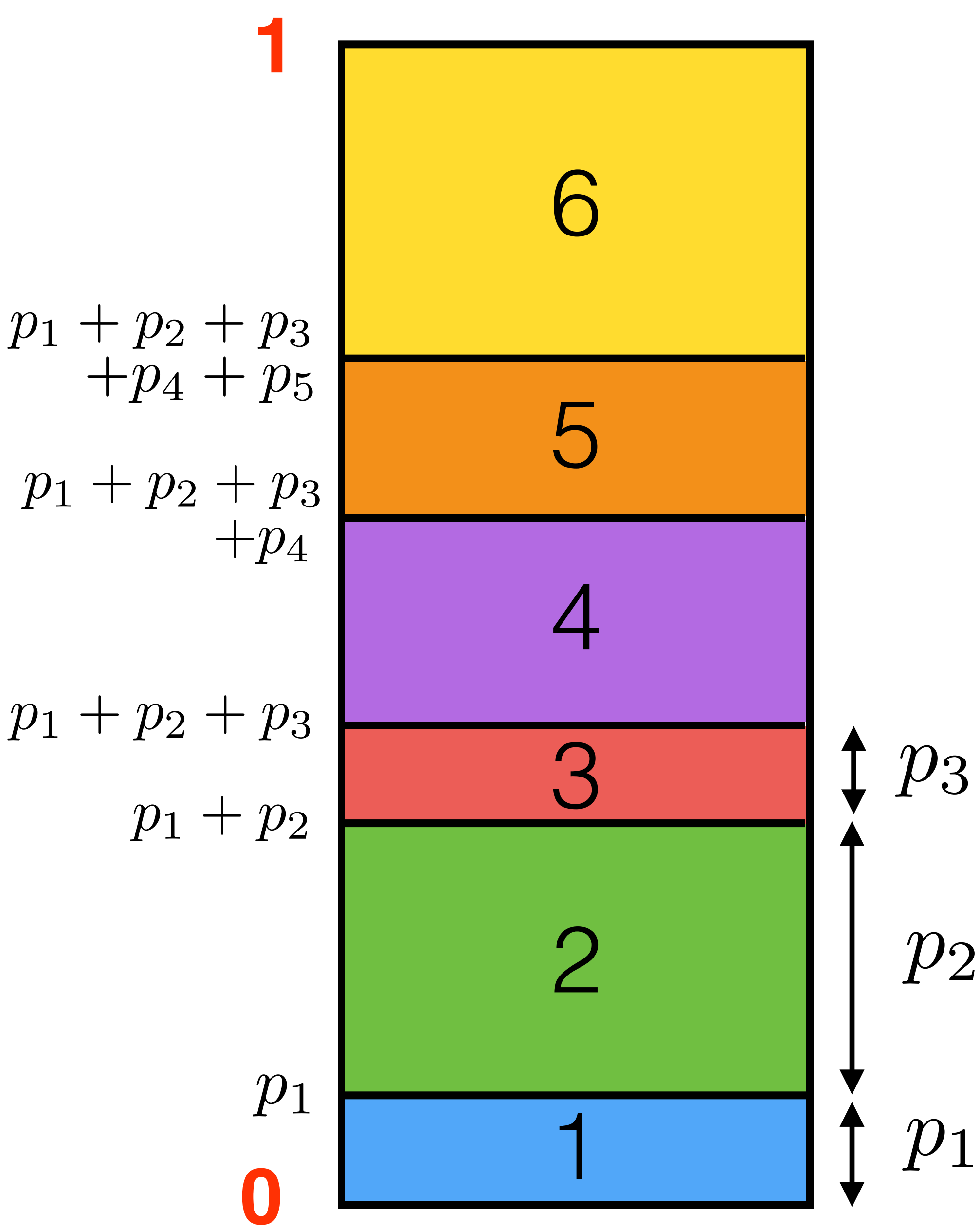
if $\epsilon < p$, then **Heads**

if $\epsilon > p$, then **Tails**

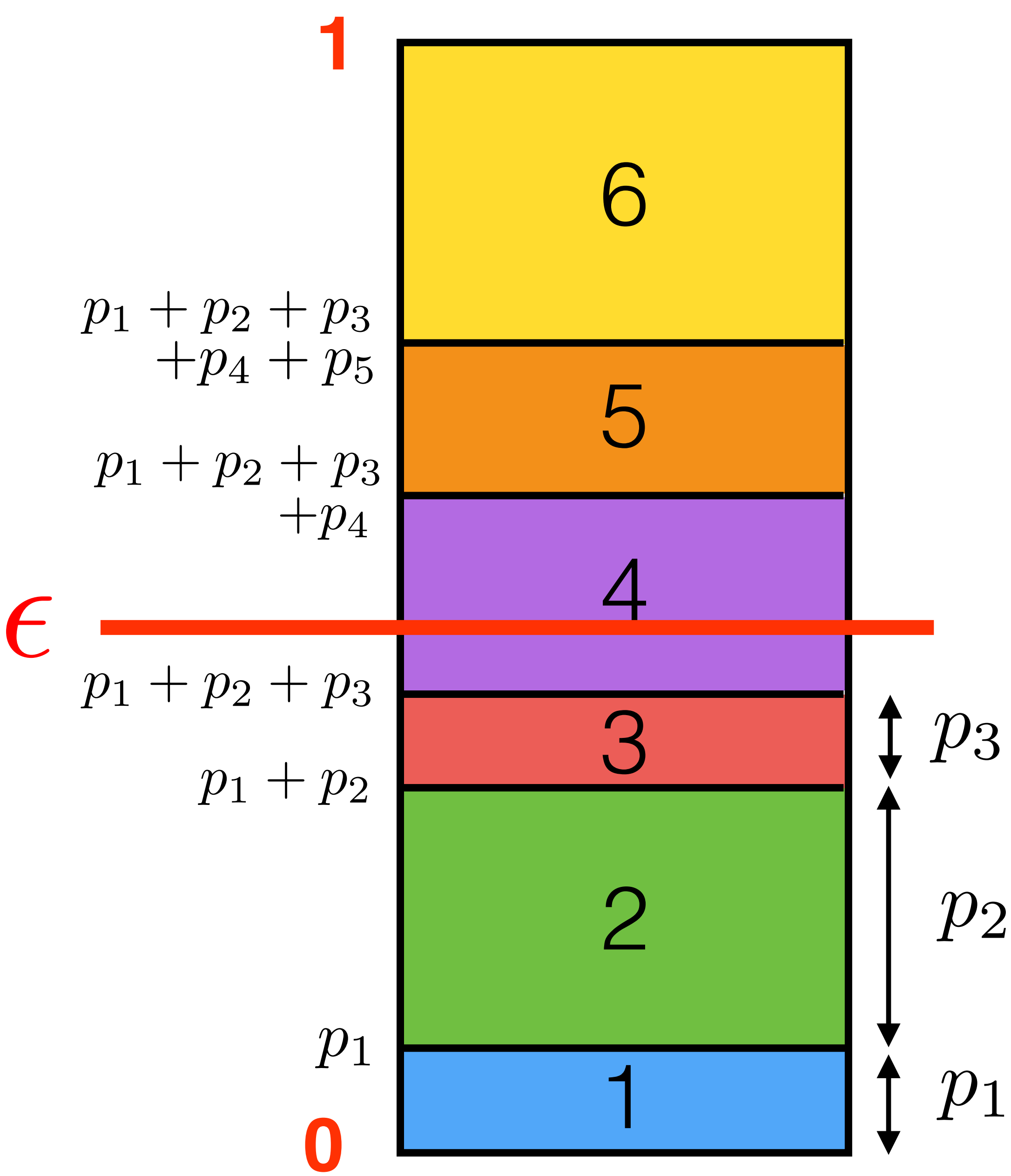
>>> Q5-6

Sampling a Discrete Random Variable

Ex. Biased dice:



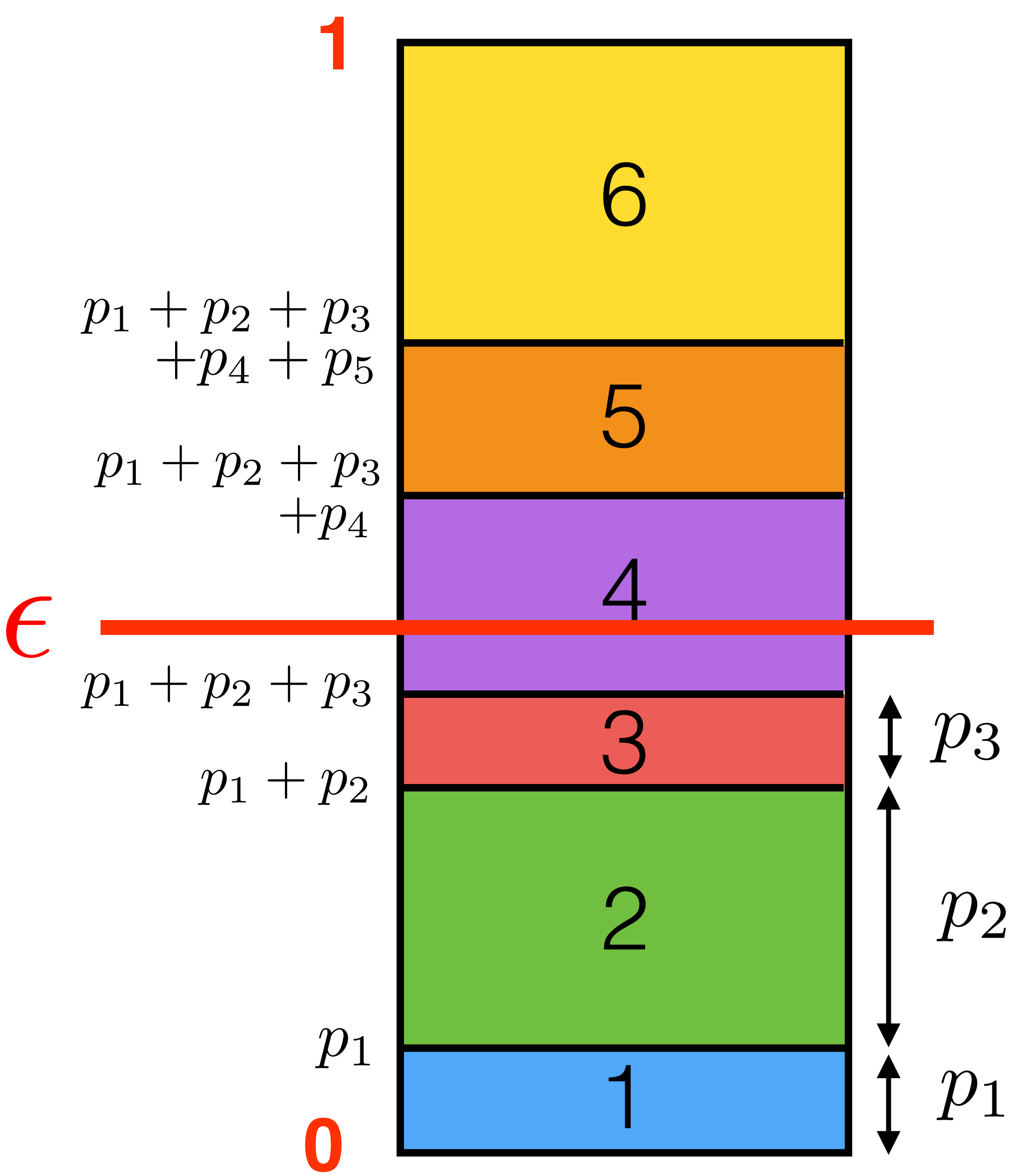
Sampling a Discrete Random Variable



Ex. Biased dice:

$\epsilon \in [0, 1)$ uniform

Sampling a Discrete Random Variable



Ex. Biased dice:

$\epsilon \in [0, 1)$ uniform

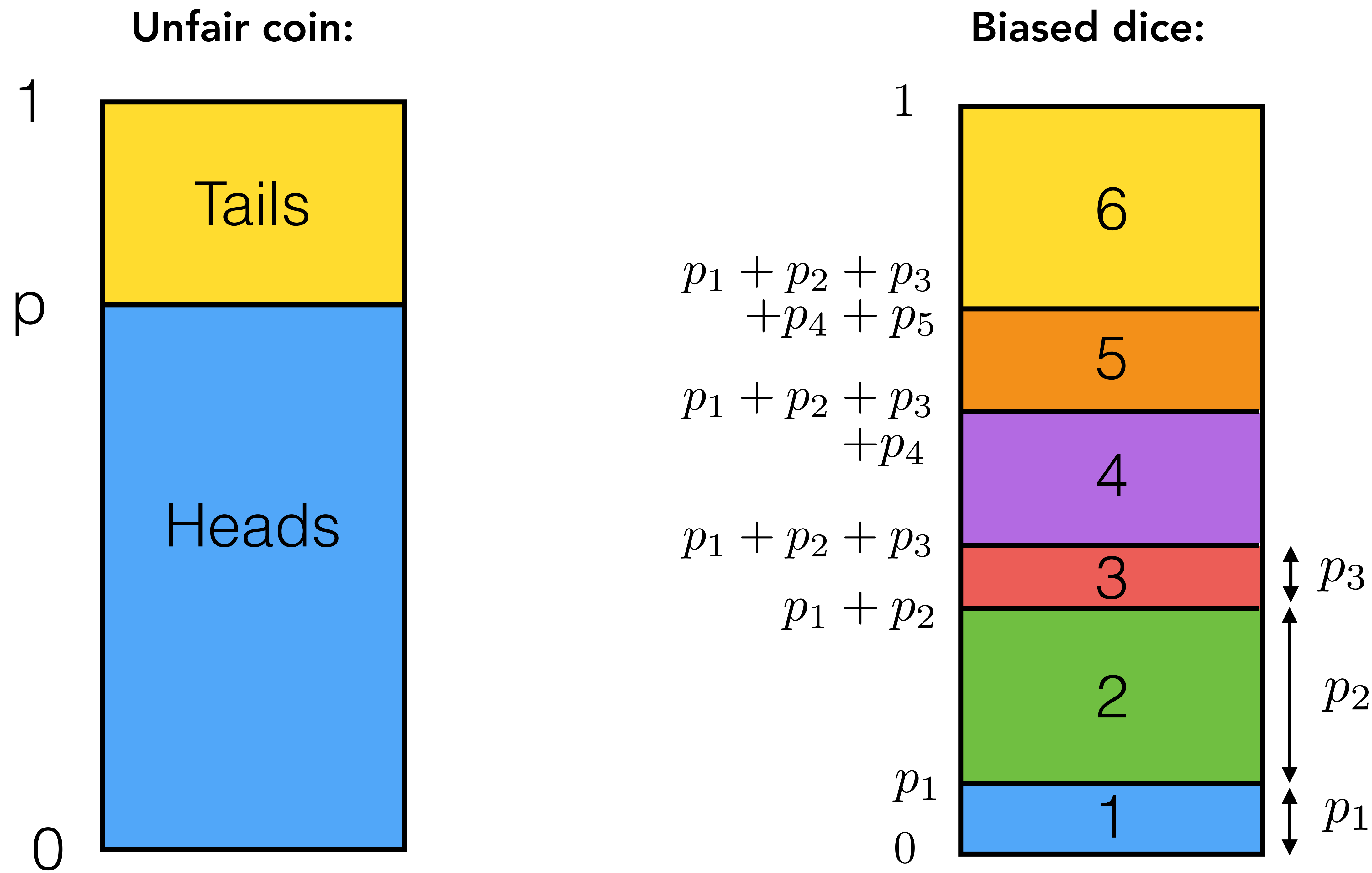
if $\epsilon < p_1$, then 1

else if $\epsilon < p_1 + p_2$, then 2

else if $\epsilon < p_1 + p_2 + p_3$, then 3

Etc.

>>> Q7-8



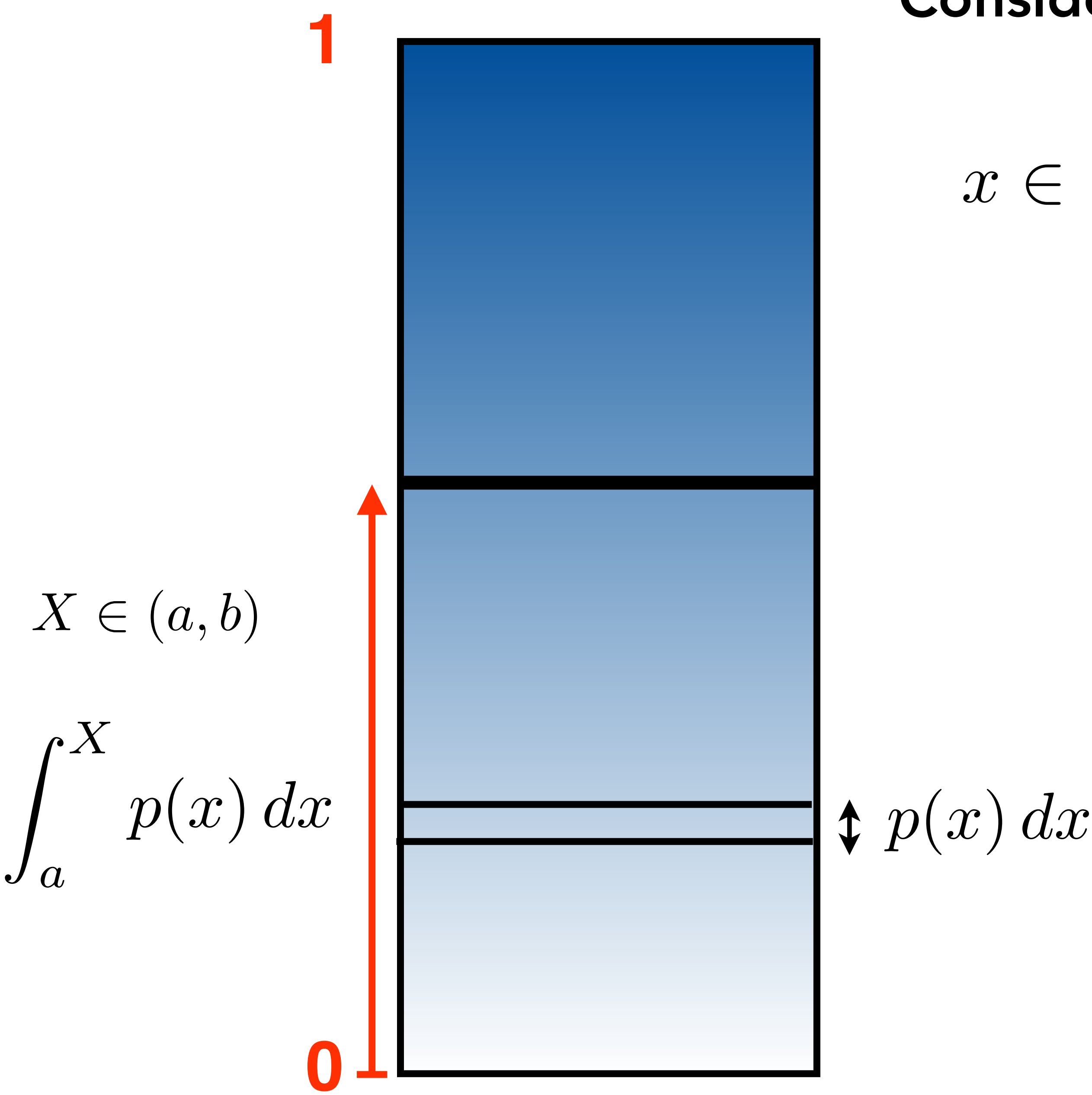
Unfair coin:
1
Tails
p
Heads
0
Biased dice:
1
6
$p_1 + p_2 + p_3$
$+p_4 + p_5$
5
$p_1 + p_2 + p_3$
$+p_4$
4
$p_1 + p_2 + p_3$
3
p_3
$p_1 + p_2$
2
p_2
p_1
1
p_1
0

Sampling a Continuous Random Variable
Using the inverse of a cumulative

Consider a continuous probability density function:

$$x \in (a, b), \qquad p(x) \qquad \text{with} \quad \int_a^b p(x) = 1$$

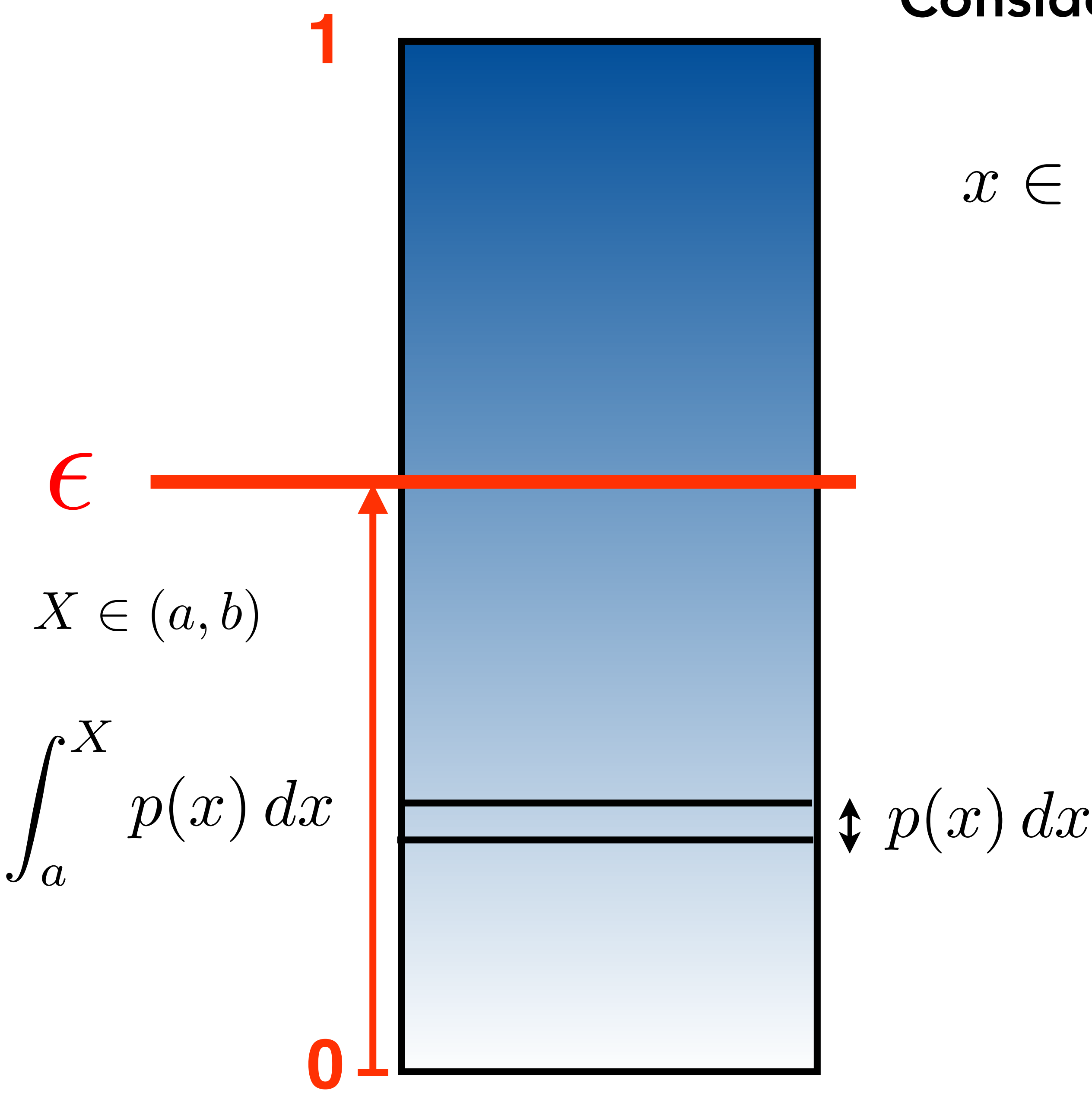
Sampling a Continuous Random Variable
Using the inverse of a cumulative



Consider a continuous probability density function:

$$x \in (a, b), \qquad p(x) \qquad \text{with} \quad \int_a^b p(x) = 1$$

Sampling a Continuous Random Variable
Using the inverse of a cumulative



Consider a continuous probability density function:

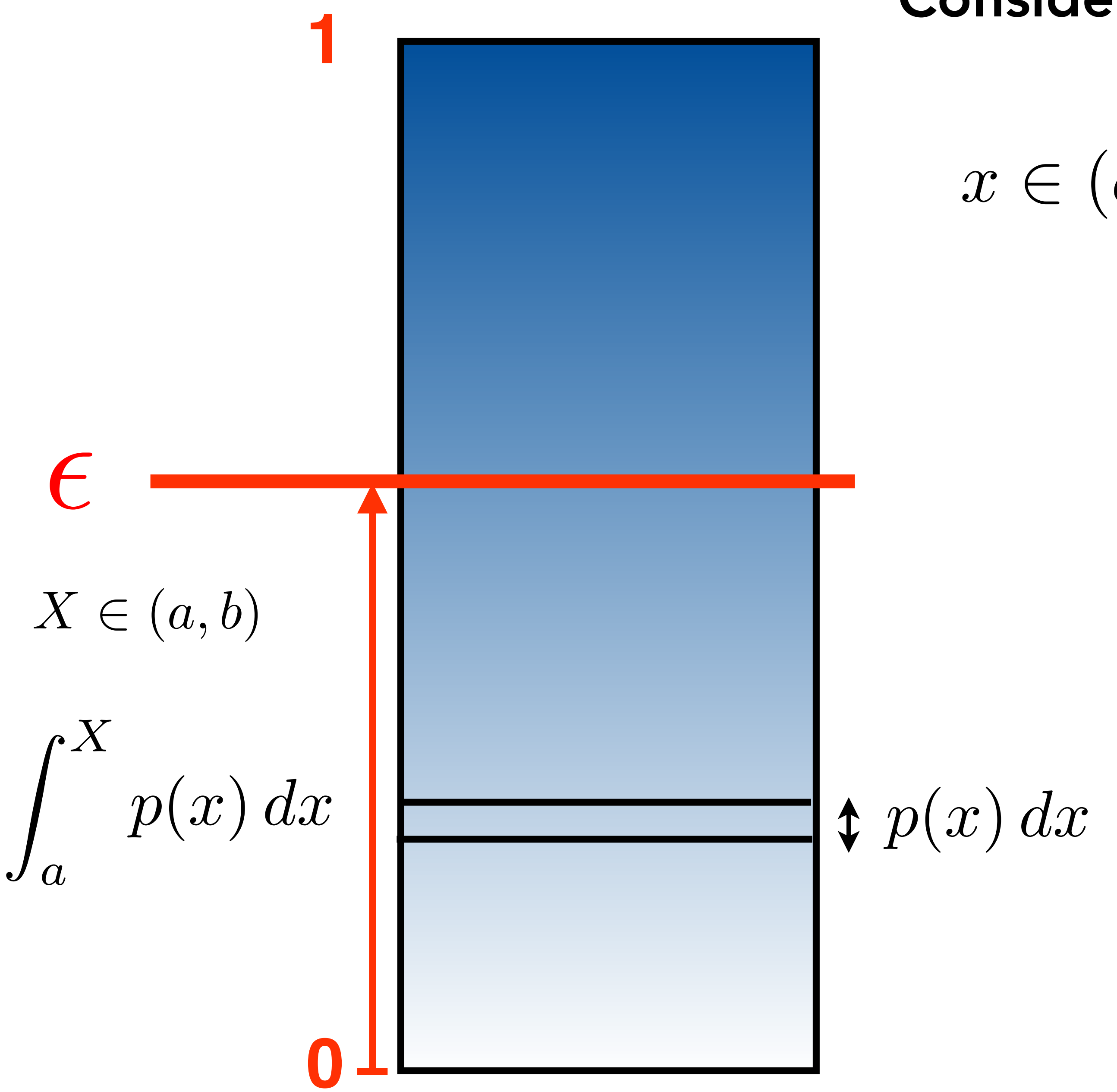
$$x \in (a,b), \qquad p(x) \qquad \text{with} \quad \int_a^b p(x) = 1$$

$$\epsilon \in [0,1) \qquad \text{uniform}$$

Find $X \in (a,b)$ **such that:**

$$\boxed{\int_a^X p(x)\,dx = \epsilon}$$

Sampling a Continuous Random Variable Using the inverse of a cumulative



Consider a continuous probability density function:

$$x \in (a,b), \qquad p(x) \qquad \text{with} \quad \int_a^b p(x) = 1$$

$$\epsilon \in [0,1) \qquad \text{uniform}$$

Find $X \in (a,b)$ **such that:**

$$\boxed{F(X) = \int_a^X p(x)\,dx} = \epsilon$$

$$\boxed{X = F^{-1}(\epsilon)}$$

Sampling a Continuous Random Variable Using the inverse of a cumulative

Ex. Exponential distribution:

Consider a series of **independent events** that happen with a **constant rate** λ

Ex. light scattering in a diffusive medium

At any time t_0, the probability that the next event happens at time $t_0 + t$ is independent of t_0 and is given by the **Exponential distribution:**

$$P(t) = \lambda \exp(-\lambda\, t)$$

- $\epsilon = \mathrm{uniform}(0, 1)$
- Find T such that: $$\int_0^T \lambda \exp(-\lambda\, t)\, dt = \epsilon$$

Sampling a Continuous Random Variable Using the inverse of a cumulative

Ex. Exponential distribution:

Consider a series of **independent events** that happen with a **constant rate** λ

Ex. light scattering in a diffusive medium

At any time t_0, the probability that the next event happens at time $t_0 + t$ is independent of t_0 and is given by the **Exponential distribution:**

$$P(t) = \lambda \exp(-\lambda\, t)$$

- $\epsilon = \mathrm{uniform}(0, 1)$
- Find T such that: $$\int_0^T \lambda \exp(-\lambda\, t)\, dt = \epsilon \quad \Rightarrow \quad \epsilon = 1 - \exp(-\lambda\, T)$$

$$\Rightarrow \quad T = -\frac{1}{\lambda} \log(1 - \epsilon)$$

Sampling a Continuous Random Variable Using the inverse of a cumulative

Ex. Exponential distribution:

Consider a series of **independent events** that happen with a **constant rate** λ

Ex. light scattering in a diffusive medium

At any time t_0 , the probability that the next event happens at time $t_0 + t$ is independent of t_0 and is given by the **Exponential distribution:**

$$P(t) = \lambda \exp(-\lambda\, t)$$

- $\epsilon = \mathrm{uniform}(0,1)$
- Find T such that: $\displaystyle\int_0^T \lambda \exp(-\lambda\, t)\, dt = \epsilon \quad \Rightarrow \quad \epsilon = 1 - \exp(-\lambda\, T)$

$$\Rightarrow \quad T = -\frac{1}{\lambda}\log(1-\epsilon) \qquad \Rightarrow \quad \boxed{T = -\frac{1}{\lambda}\log(\eta) \quad \text{where } \eta = \mathrm{uniform}(0,1)}$$

Sampling a Continuous Random Variable Using the inverse of a cumulative

Ex. Exponential distribution:

Consider a series of **independent events** that happen with a **constant rate** λ

Ex. light scattering in a diffusive medium

At any time t_0, the probability that the next event happens at time $t_0 + t$ is independent of t_0 and is given by the **Exponential distribution:**

$$P(t) = \lambda \exp(-\lambda\, t)$$

>>> Q9

>>> Q10-11

- $\epsilon = \text{uniform}(0,1)$
- Find T such that: $$\int_0^T \lambda \exp(-\lambda\, t)\, dt = \epsilon \quad \Rightarrow \quad \epsilon = 1 - \exp(-\lambda\, T)$$

$$\Rightarrow \quad T = -\frac{1}{\lambda} \log(1-\epsilon) \quad \Rightarrow \quad \boxed{T = -\frac{1}{\lambda} \log(\eta) \quad \text{where } \eta = \text{uniform}(0,1)}$$

Part 3

Monte Carlo Simulation

Book. *Algorithms and Computations*, by Werner Krauth

Introduction to Monte Carlo

Ex. Calculating the area of a circle:

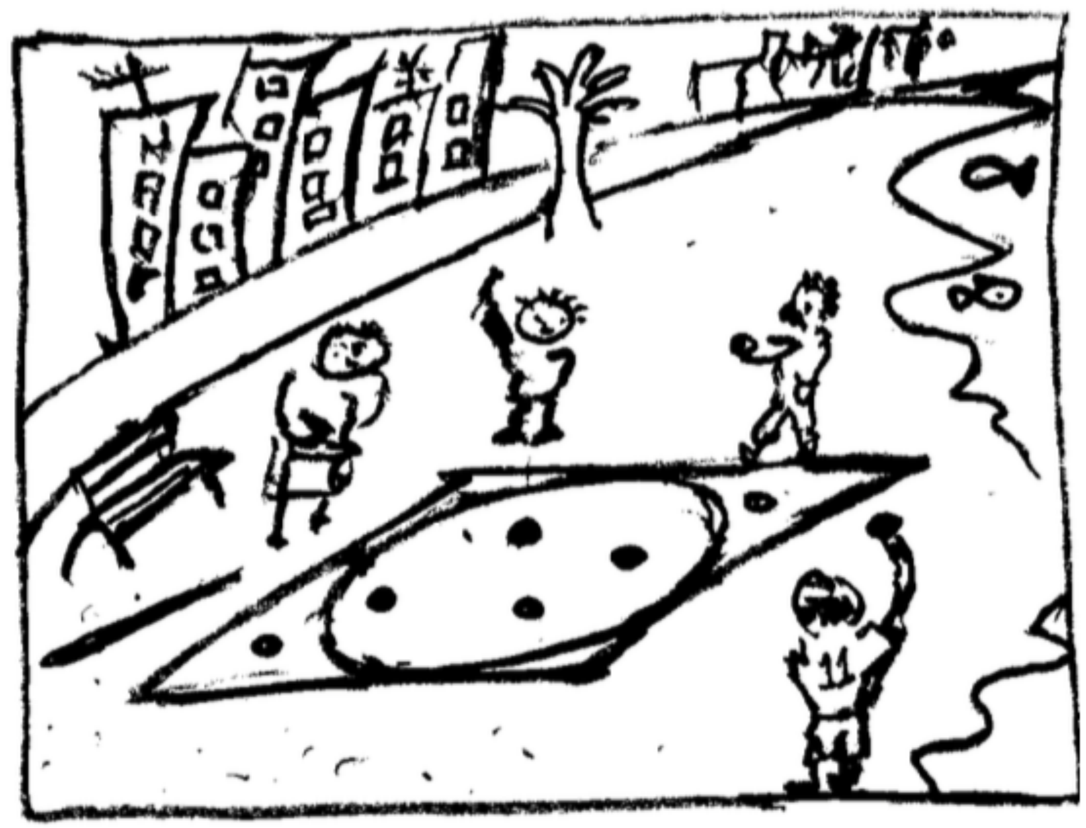
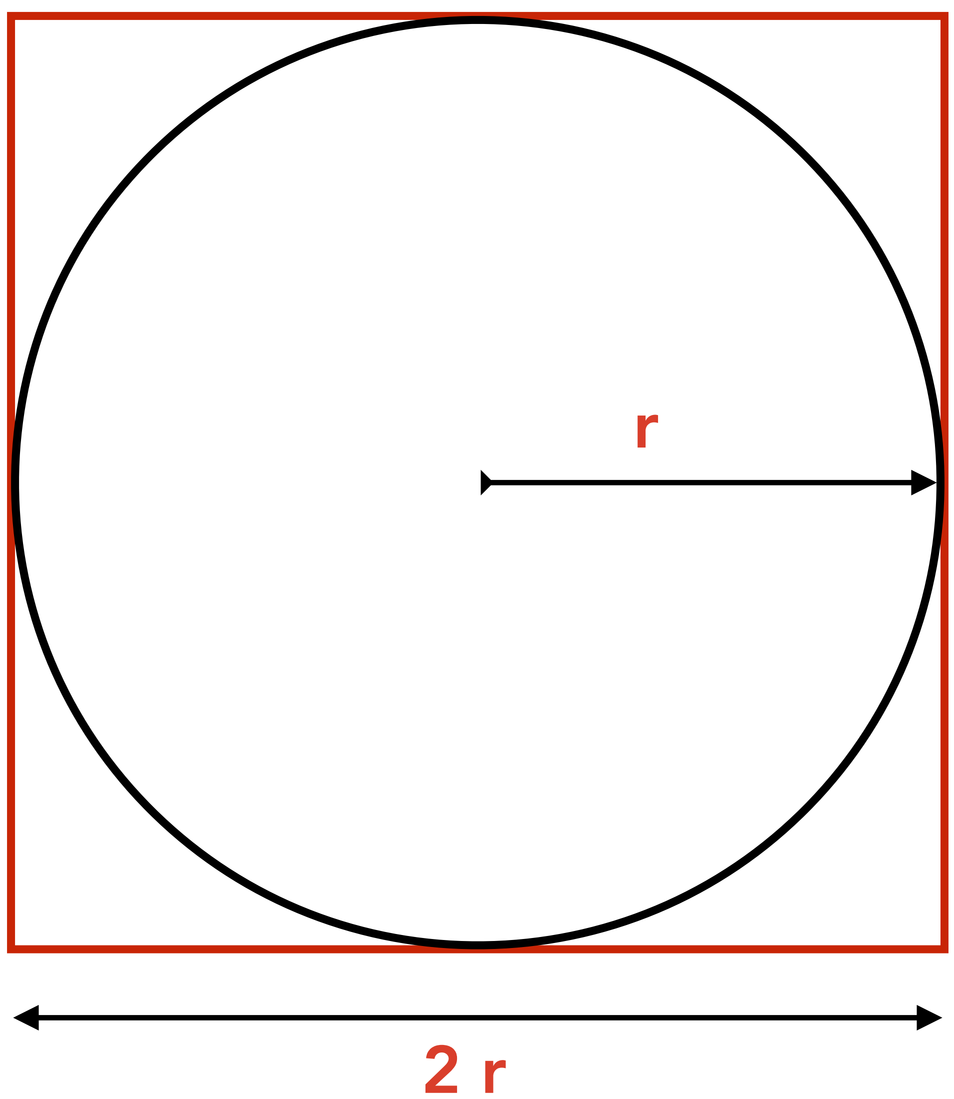


Fig. 1.1 Children computing the number π on the Monte Carlo beach.

Introduction to Monte Carlo

Ex. Calculating the area of a circle:

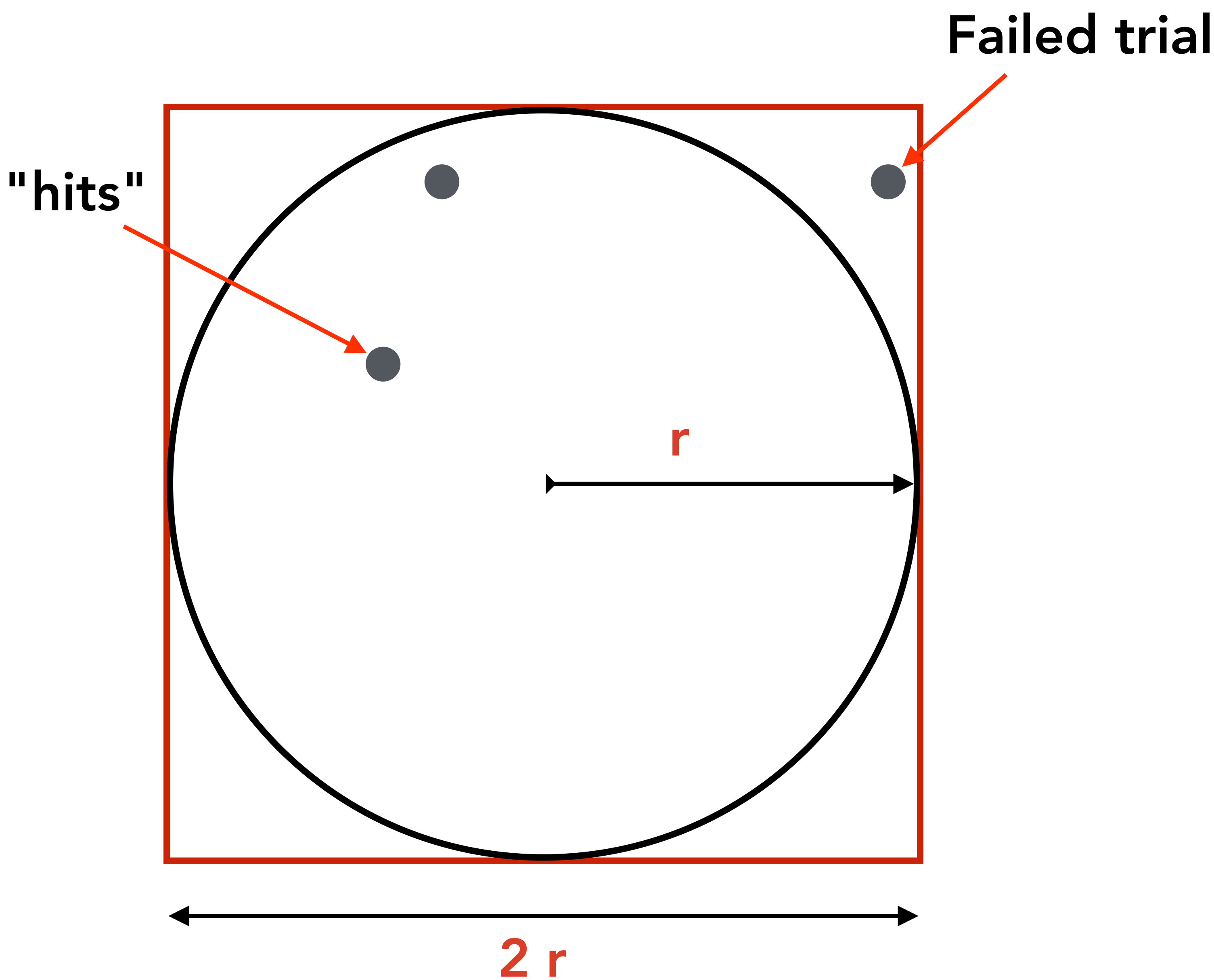
$$\frac{\text{area circle}}{\text{area square}} = \frac{\pi\, r^2}{(2\, r)^2} = \frac{\pi}{4}$$



Introduction to Monte Carlo

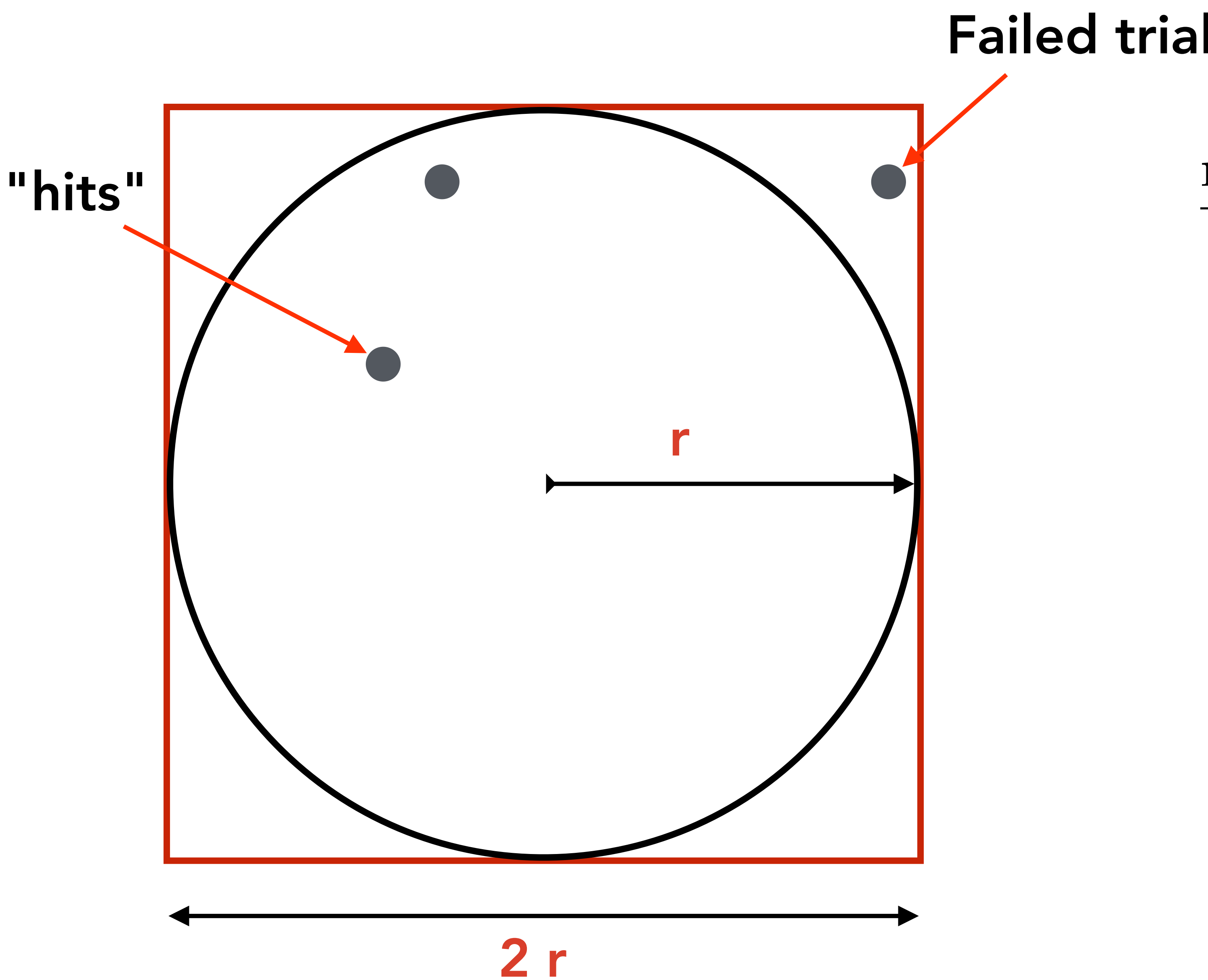
Ex. Calculating the area of a circle:

$$\frac{\text{area circle}}{\text{area square}} = \frac{\pi\, r^2}{(2\, r)^2} = \frac{\pi}{4}$$



Introduction to Monte Carlo

Ex. Calculating the area of a circle:



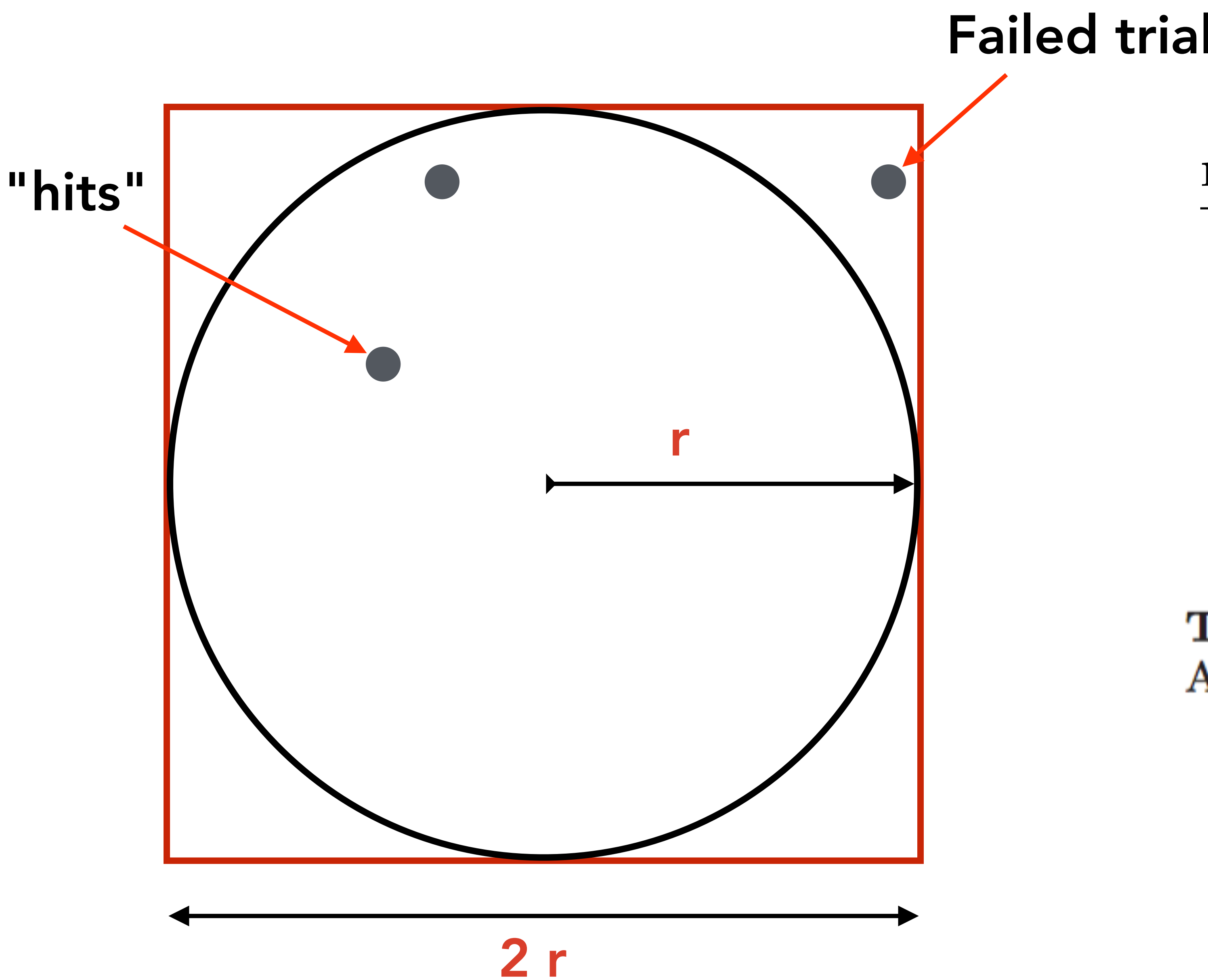
$$\frac{\text{area circle}}{\text{area square}} = \frac{\pi\, r^2}{(2\, r)^2} = \frac{\pi}{4}$$

$$\frac{\text{number of "hits"}}{\text{number of trials}} \simeq \frac{\text{area circle}}{\text{area square}}$$

$$\boxed{\pi \simeq 4\,\frac{\text{number of "hits"}}{\text{number of trials}}}$$

Introduction to Monte Carlo

Ex. Calculating the area of a circle:



$$\frac{\text{area circle}}{\text{area square}} = \frac{\pi\, r^2}{(2\, r)^2} = \frac{\pi}{4}$$

$$\frac{\text{number of "hits"}}{\text{number of trials}} \simeq \frac{\text{area circle}}{\text{area square}}$$

$$\boxed{\pi \simeq 4\, \frac{\text{number of "hits"}}{\text{number of trials}}}$$

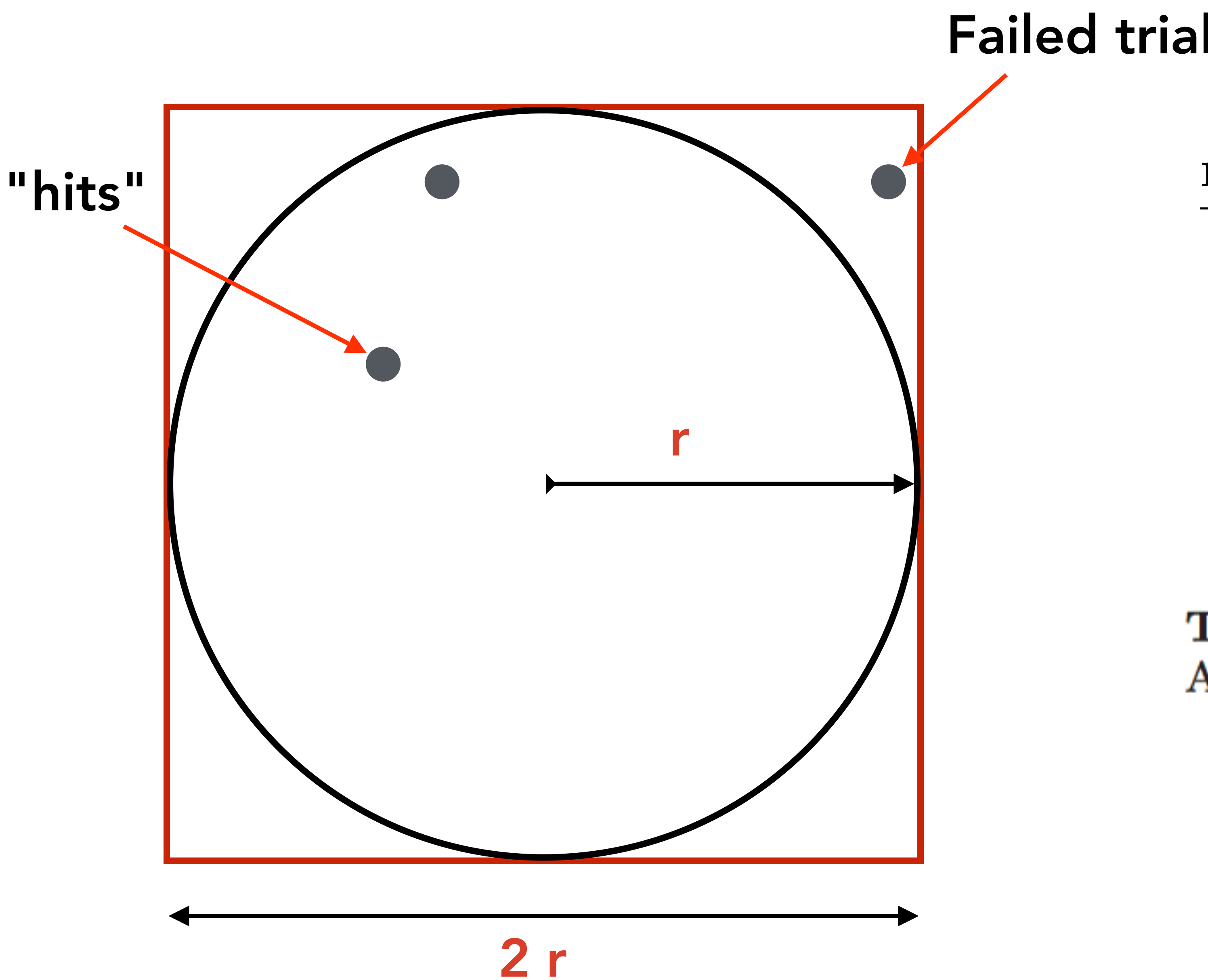
Table 1.1 Results of five runs of Alg. 1.1 (`direct-pi`) with $N = 4000$

Run	N_{hits}	Estimate of π
1	3156	3.156
2	3150	3.150
3	3127	3.127
4	3171	3.171
5	3148	3.148

>>> Q13-14

Introduction to Monte Carlo

Ex. Calculating the area of a circle:



$$\frac{\text{area circle}}{\text{area square}} = \frac{\pi\, r^2}{(2\, r)^2} = \frac{\pi}{4}$$

$$\frac{\text{number of "hits"}}{\text{number of trials}} \simeq \frac{\text{area circle}}{\text{area square}}$$

$$\boxed{\pi \simeq 4\, \frac{\text{number of "hits"}}{\text{number of trials}}}$$

Table 1.1 Results of five runs of Alg. 1.1 (`direct-pi`) with $N = 4000$

Run	N_{hits}	Estimate of π
1	3156	3.156
2	3150	3.150
3	3127	3.127
4	3171	3.171
5	3148	3.148

Powerful approach for calculating integrals!

Going further in learning computational methods
Possible interesting courses

Numerical Algorithms (6EC block 2). Link

Solving eigenvalue problems, non-linear equations, optimization problems, interpolations, etc.

Stochastic Simulation (6EC block 2). Link

Including, Statistical analysis of data, hypothesis testing,
Monte Carlo, Importance sampling, simulated annealing