CSE 312

Foundations of Computing II

Lecture 14: Wrapup of Bloom filters + Continuous RV

Agenda

- Wrap-up of Bloom Filters
- Continuous Random Variables
- Probability Density Function
- Cumulative Distribution Function
- Expectation and Variance of continuous RVs

Bloom Filters – Main points

- Probabilistic data structure.
- Close cousins of hash tables.
  - But: Ridiculously space efficient
- Occasional errors, specifically false positives.

Bloom Filters

- Stores information about a set of elements $S \subseteq U$.
- Supports two operations:
  1. **`add`**(x) - adds $x \in U$ to the set S
  2. **`contains`**(x) – ideally: true if $x \in S$, false otherwise

Instead, relaxed guarantees:

- False $\rightarrow$ **definitely** not in S
- True $\rightarrow$ **possibly** in S

  [i.e. we could have *false positives*]

Bloom Filters – Why Accept False Positives?

- **Speed** – both `add` and `contains` very very fast.
- **Space** – requires a miniscule amount of space relative to storing all the actual items that have been added.
  - Often just 8 bits per inserted item!
- **Fallback mechanism** – can distinguish false positives from true positives with extra cost
  - Ok if mostly negatives expected + low false positive rate

Bloom Filters – Ingredients

Basic data structure is a $k \times m$ binary array "the Bloom filter"

- k rows $t_1, \ldots, t_k$, each of size m
- Think of each row as an m-bit vector

k different hash functions $\mathbf{h}_1, \ldots, \mathbf{h}_k : U \rightarrow [m]$

Bloom Filters – Three operations

- Set up Bloom filter for $S = \emptyset$

function INITIALIZE(k, m)
  for $i = 1, \ldots, k$: **do**
    t_i = new bit vector of m 0s

- Update Bloom filter for $S \leftarrow S \cup \{x\}$

function ADD(x)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

- Check if $x \in S$

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function INITIALIZE(k, m)
for $i = 1, \ldots, k$: **do**
t_i = new bit vector of m 0s

Index →	0	1	2	3	4
t_1	0	0	0	0	0
t_2	0	0	0	0	0
t_3	0	0	0	0	0

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function ADD(x)
  for $i = 1, \dots, k$: **do**
    $t_i[h_i(x)] = 1$

add("thisisavirus.com")

h_1("thisisavirus.com") $\rightarrow$ 2

Index $\rightarrow$	0	1	2	3	4
t_1	0	0	0	0	0
t_2	0	0	0	0	0
t_3	0	0	0	0	0

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function ADD(x)
for $i = 1, \ldots, k$: **do**
$t_i[h_i(x)] = 1$

add("thisisavirus.com")

h_1("thisisavirus.com") → 2

h_2("thisisavirus.com") → 1

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	0	0	0	0
t_3	0	0	0	0	0

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function ADD(x)
  for $i = 1, \ldots, k$: **do**
    $t_i[h_i(x)] = 1$

add("thisisavirus.com")

h_1("thisisavirus.com") → 2
h_2("thisisavirus.com") → 1
h_3("thisisavirus.com") → 4

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	0

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function ADD(x)
for $i = 1, \ldots, k$: **do**
$t_i[h_i(x)] = 1$

add("thisisavirus.com")

h_1("thisisavirus.com") → 2

h_2("thisisavirus.com") → 1

h_3("thisisavirus.com") → 4

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Contains

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

Returns True if the bit vector t_i for each hash function has bit 1 at index determined by $h_i(x)$,
Returns False otherwise

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
    return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

contains("thisisavirus.com")

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
    return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True

contains("thisisavirus.com")

h_1("thisisavirus.com") $\rightarrow$ 2

Index $\rightarrow$	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
    return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True   True

contains("thisisavirus.com")

h_1("thisisavirus.com") → 2

h_2("thisisavirus.com") → 1

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True True True

contains("thisisavirus.com")

h_1("thisisavirus.com") → 2

h_2("thisisavirus.com") → 1

h_3("thisisavirus.com") → 4

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: Example

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \land t_2[h_2(x)] == 1 \land \cdots \land t_k[h_k(x)] == 1$

True True True

contains("thisisavirus.com")

h_1("thisisavirus.com") $\rightarrow$ 2

h_2("thisisavirus.com") $\rightarrow$ 1

h_3("thisisavirus.com") $\rightarrow$ 4

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Since all conditions satisfied, returns True (correctly)

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

add("totallynotsuspicious.com")

function ADD(x)
    for $i = 1, \ldots, k$: **do**
        $t_i[h_i(x)] = 1$

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function ADD(x)
  for $i = 1, \dots, k$: **do**
    $t_i[h_i(x)] = 1$

add("totallynotsuspicious.com")

h_1("totallynotsuspicious.com") → 1

Index →	0	1	2	3	4
t_1	0	0	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function ADD(x)
    for $i = 1, \dots, k$: **do**
        $t_i[h_i(x)] = 1$

add("totallynotsuspicious.com")

h_1("totallynotsuspicious.com") → 1

h_2("totallynotsuspicious.com") → 0

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	0	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

$$\textbf{function } \text{ADD}(x)$$
$$\textbf{for } i = 1, \ldots, k \textbf{: do}$$
$$t_i[h_i(x)] = 1$$

add("totallynotsuspicious.com")

h_1("totallynotsuspicious.com") → 1

h_2("totallynotsuspicious.com") → 0

h_3("totallynotsuspicious.com") → 4

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

$$\textbf{function } \text{ADD}(x)$$
$$\textbf{for } i = 1, \ldots, k\textbf{: do}$$
$$t_i[h_i(x)] = 1$$

add(“totallynotsuspicious.com”)

h_1(“totallynotsuspicious.com”) → 1

h_2(“totallynotsuspicious.com”) → 0

h_3(“totallynotsuspicious.com”) → 4

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

contains("verynormalsite.com")

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True

contains("verynormalsite.com")

h_1("verynormalsite.com") → 2

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
    return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True   True

contains("verynormalsite.com")

h_1("verynormalsite.com") → 2

h_2("verynormalsite.com") → 0

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \wedge t_2[h_2(x)] == 1 \wedge \cdots \wedge t_k[h_k(x)] == 1$

True True True

contains("verynormalsite.com")

h_1("verynormalsite.com") → 2

h_2("verynormalsite.com") → 0

h_3("verynormalsite.com") → 4

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Bloom Filters: False Positives

Bloom filter t of length $\boldsymbol{m}$ = 5 that uses $\boldsymbol{k}$ = 3 hash functions

function CONTAINS(x)
  return $t_1[h_1(x)] == 1 \land t_2[h_2(x)] == 1 \land \cdots \land t_k[h_k(x)] == 1$

True True True

contains("verynormalsite.com")

h_1("verynormalsite.com") → 2

h_2("verynormalsite.com") → 0

h_3("verynormalsite.com") → 4

Index →	0	1	2	3	4
t_1	0	1	1	0	0
t_2	1	1	0	0	0
t_3	0	0	0	0	1

Since all conditions satisfied, returns True (incorrectly)

Analysis: False positive probability

Question: For an element $x \in U$, what is the probability that **`contains`**(x) returns true if **`add`**(x) was never executed before?

Probability over what?! Over the choice of the $\boldsymbol{h}_1, \dots, \boldsymbol{h}_k$

Assumptions for the analysis:

- Each $\mathbf{h}_i(x)$ is uniformly distributed in $[m]$ for all x and i
- Hash function outputs for each $\mathbf{h}_i$ are mutually independent (not just in pairs)
- Different hash functions are independent of each other

False positive probability – Events

Assume we perform $\texttt{add}(x_1), \ldots, \texttt{add}(x_n)$
+ $\texttt{contains}(x)$ for $x \notin \{x_1, \ldots, x_n\}$

False positive iff $\mathbf{h}_i(x) \in \{\mathbf{h}_i(x_1), \ldots, \mathbf{h}_i(x_n)\}$. $\quad \forall i = 1, \ldots, k$

$P(\text{false positive})$

False positive probability – Events

Assume we perform **add**$(x_1), \ldots,$**add**(x_n)
+ **contains**(x) for $x \notin \{x_1, \ldots, x_n\}$

Event E_i holds iff $\mathbf{h}_i(x) \in \{\mathbf{h}_i(x_1), \ldots, \mathbf{h}_i(x_n)\}$

$$P(\text{false positive}) = P(E_1 \cap E_2 \cap \cdots \cap E_k) = \prod_{i=1}^{k} P(E_i)$$

$\mathbf{h}_1, \ldots, \mathbf{h}_k$ independent

False positive probability – Events

Event E_i holds iff $\mathbf{h}_i(x) \in \{\mathbf{h}_i(x_1), \ldots, \mathbf{h}_i(x_n)\}$

Event E_i^c holds iff $\mathbf{h}_i(x) \neq \mathbf{h}_i(x_1)$ and … and $\mathbf{h}_i(x) \neq \mathbf{h}_i(x_n)$

$$P(E_i^c) = \sum_{z=1}^{m} P(\mathbf{h}_i(x) = z) \cdot P(E_i^c \mid \mathbf{h}_i(x) = z)$$

LTP

False positive probability – Events

Event E_i^c holds iff $\mathbf{h}_i(x) \neq \mathbf{h}_i(x_1)$ and … and $\mathbf{h}_i(x) \neq \mathbf{h}_i(x_n)$

$$P(E_i^c \mid \mathbf{h}_i(x) = z) = P(\mathbf{h}_i(x_1) \neq z, \dots, \mathbf{h}_i(x_n) \neq z \mid \mathbf{h}_i(x) = z)$$

$$= P(\mathbf{h}_i(x_1) \neq z, \dots, \mathbf{h}_i(x_n) \neq z)$$

$$= \prod_{j=1}^{n} P(\mathbf{h}_i(x_j) \neq z)$$

Independence of values of $\boldsymbol{h}_i$ on different inputs

False positive probability – Events

Event E_i^c holds iff $\mathbf{h}_i(x) \neq \mathbf{h}_i(x_1)$ and … and $\mathbf{h}_i(x) \neq \mathbf{h}_i(x_n)$

$$P(E_i^c \mid \mathbf{h}_i(x) = z) = P(\mathbf{h}_i(x_1) \neq z, \dots, \mathbf{h}_i(x_n) \neq z \mid \mathbf{h}_i(x) = z)$$

Independence of values of $\boldsymbol{h}_i$ on different inputs

$$= P(\mathbf{h}_i(x_1) \neq z, \dots, \mathbf{h}_i(x_n) \neq z\,)$$

$$= \prod_{j=1}^{n} P(\mathbf{h}_i(x_j) \neq z)$$

Outputs of $\boldsymbol{h}_i$ uniformly spread

$$= \prod_{j=1}^{n} \left(1 - \frac{1}{m}\right) = \left(1 - \frac{1}{m}\right)^n$$

$$\Rightarrow\ P(E_i^c) = \sum_{z=1}^{m} P(\mathbf{h}_i(x) = z) \cdot P(E_i^c \mid \mathbf{h}_i(x) = z) = \left(1 - \frac{1}{m}\right)^n$$

False positive probability – Events

Event E_i holds iff $\mathbf{h}_i(x) \in \{\mathbf{h}_i(x_1), \dots, \mathbf{h}_i(x_n)\}$

Event E_i^c holds iff $\mathbf{h}_i(x) \neq \mathbf{h}_i(x_1)$ and … and $\mathbf{h}_i(x) \neq \mathbf{h}_i(x_n)$

$$P(E_i^c) = \left(1 - \frac{1}{m}\right)^n$$

$$\Rightarrow \quad \text{FPR} = \prod_{i=1}^{k}\left(1 - P(E_i^c)\right) = \left(1 - \left(1 - \frac{1}{m}\right)^n\right)^k$$

False Positivity Rate – Example

$$\text{FPR} = \left(1 - \left(1 - \frac{1}{m}\right)^{n}\right)^{k}$$

e.g., $n = 5{,}000{,}000$

$k = 30$

$m = 2{,}500{,}000$

FPR = 1.28%

Comparison with Hash Tables - Space

- Google storing 5 million URLs, each URL 40 bytes.
- Bloom filter with $k = 30$ and $m = 2{,}500{,}000$

Hash Table

(optimistic)
$5{,}000{,}000 \times 40B = 200\text{MB}$

Bloom Filter

$2{,}500{,}000 \times 30 = 75{,}000{,}000$ bits

< 10 MB

Time

- Say avg user visits 102,000 URLs in a year, of which 2,000 are malicious.
- 0.5 seconds to do lookup in the database, 1ms for lookup in Bloom filter.
- Suppose the false positive rate is 3%

$$1\text{ms} + \frac{100000 \times 0.03 \times 500\text{ms} \quad + 2000 \times 500 \text{ ms}}{102000} \approx 25.51\text{ms}$$

0.5 seconds DB lookup

false positives

malicious URLs

total URLs

Bloom filter lookup

Bloom Filters typical of….

… randomized algorithms and randomized data structures.

- **Simple**
- **Fast**
- **Efficient**
- **Elegant**
- **Useful!**

Agenda

- Wrap-up of Bloom Filters
- Continuous Random Variables
- Probability Density Function
- Cumulative Distribution Function
- Expectation and Variance of continuous r.v.

Often we want to model experiments where the outcome is not discrete.

Hope you enjoyed the zoo! 🦍🐘🦁🐅🦓🐪🦒

$X \sim \text{Unif}(a, b)$

$$P(X = k) = \frac{1}{b - a + 1}$$

$$\mathbb{E}[X] = \frac{a + b}{2}$$

$$\text{Var}(X) = \frac{(b - a)(b - a + 2)}{12}$$

$X \sim \text{Ber}(p)$

$$P(X = 1) = p, P(X = 0) = 1 - p$$

$$\mathbb{E}[X] = p$$

$$\text{Var}(X) = p(1 - p)$$

$X \sim \text{Bin}(n, p)$

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

$$\mathbb{E}[X] = np$$

$$\text{Var}(X) = np(1 - p)$$

$X \sim \text{Geo}(p)$

$$P(X = k) = (1 - p)^{k-1} p$$

$$\mathbb{E}[X] = \frac{1}{p}$$

$$\text{Var}(X) = \frac{1 - p}{p^2}$$

$X \sim \text{NegBin}(r, p)$

$$P(X = k) = \binom{k - 1}{r - \ldots}$$

$$\mathbb{E}[X] = \frac{r}{p}$$

$$\text{Var}(X) = \frac{r(1 - \ldots}{p^2}$$

$X \sim \text{HypGeo}(N, K, n)$

$$\frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

$$\frac{\ldots(N - K)(N - n)}{N^2(N - 1)}$$

$X \sim \text{Poisson}(\lambda)$

$$P(X = i) = e^{-\lambda} \cdot \frac{\lambda^i}{i!}$$

$$E[X] = \lambda$$

$$\text{Var}(X) = \lambda$$

Agenda

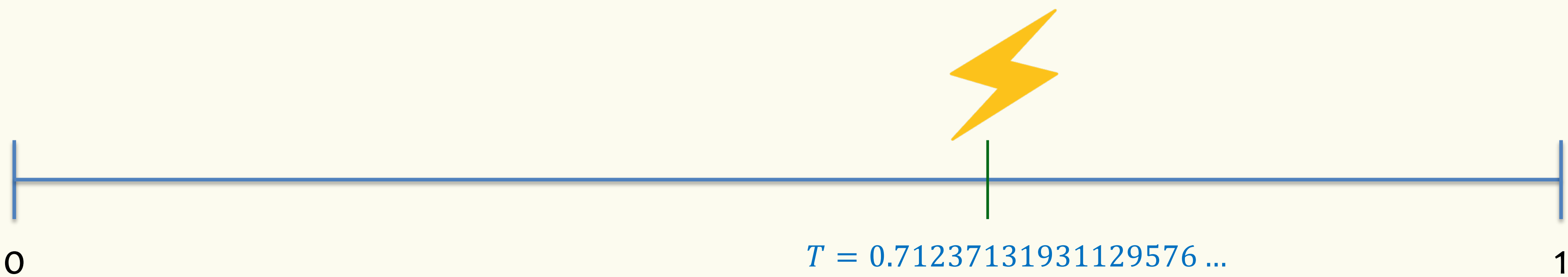
- Wrap-up of Bloom Filters
- Continuous Random Variables ◀
- Probability Density Function
- Cumulative Distribution Function
- Expectation and Variance of continuous r.v.

Often we want to model experiments where the outcome is not discrete.

Example – Lightning Strike

Lightning strikes a pole within a one-minute time frame

- T = time of lightning strike
- Every time within $[0,1]$ is equally likely
  - Time measured with infinitesimal precision.

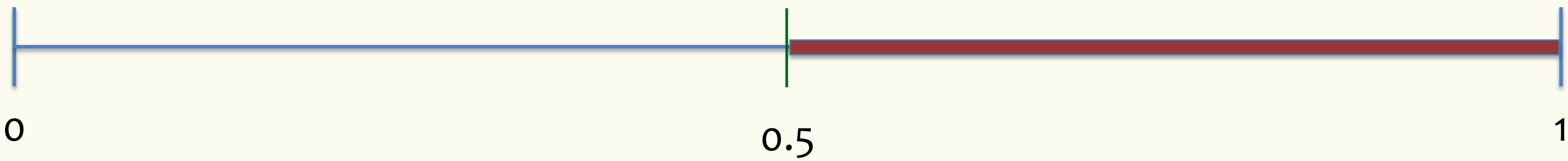


The outcome space is not discrete

Lightning strikes a pole within a one-minute time frame

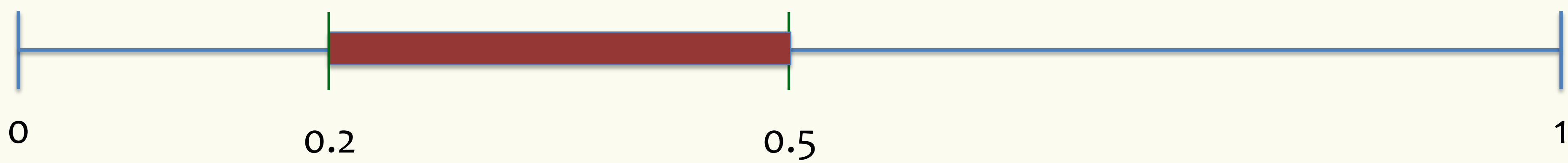
- T = time of lightning strike
- Every point in time within $[0,1]$ is equally likely

$P(T \geq 0.5) =$



Lightning strikes a pole within a one-minute time frame

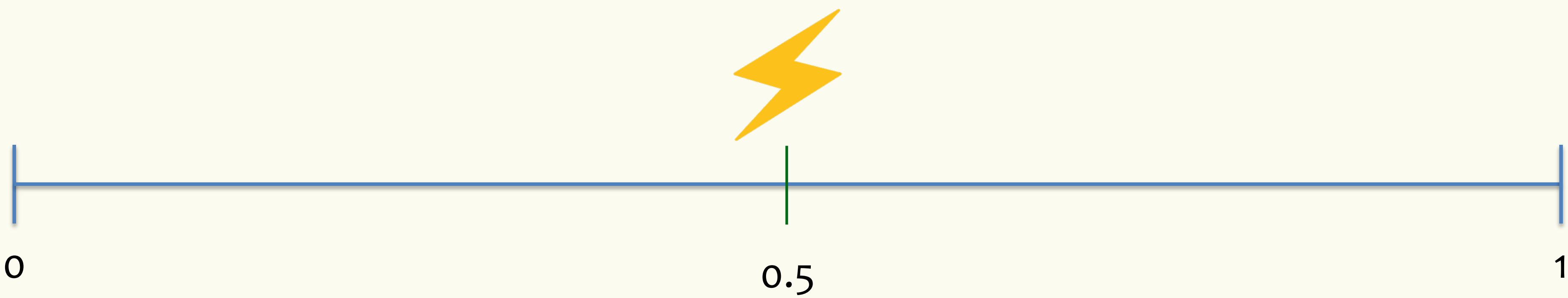
- T = time of lightning strike
- Every point in time within $[0,1]$ is equally likely



$P(0.2 \leq T \leq 0.5) =$

Lightning strikes a pole within a one-minute time frame

- T = time of lightning strike
- Every point in time within $[0,1]$ is equally likely



$P(T = 0.5) =$

Bottom line

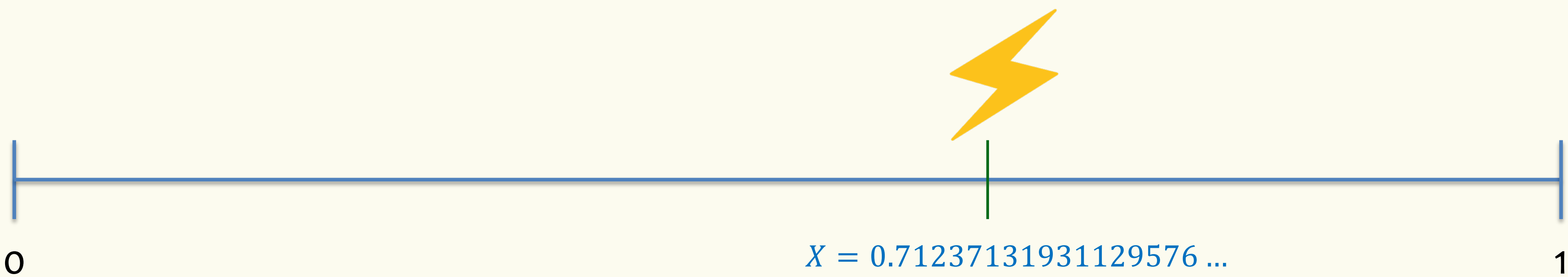
- This gives rise to a different type of random variable
- $P(T = x) = 0$ for all $x \in [0,1]$
- Yet, somehow we want
  - $P(T \in [0,1]) = 1$
  - $P(T \in [a, b]) = b - a$
  - …
- How do we model the behavior of T?

First try: A discrete approximation

Example – Lightning Strike

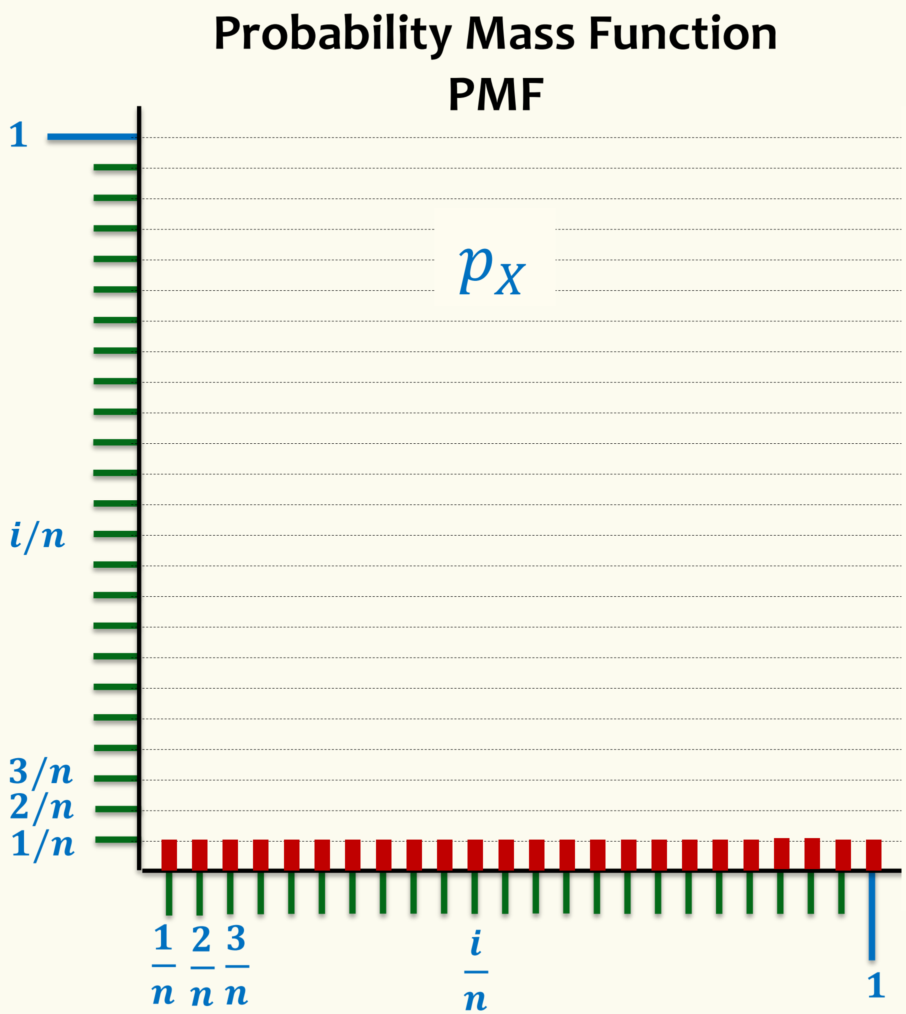
Lightning strikes a pole within a one-minute time frame

- X = time of lightning strike
- Every time within $[0,1]$ is equally likely
  - – Time measured with infinitesimal precision.

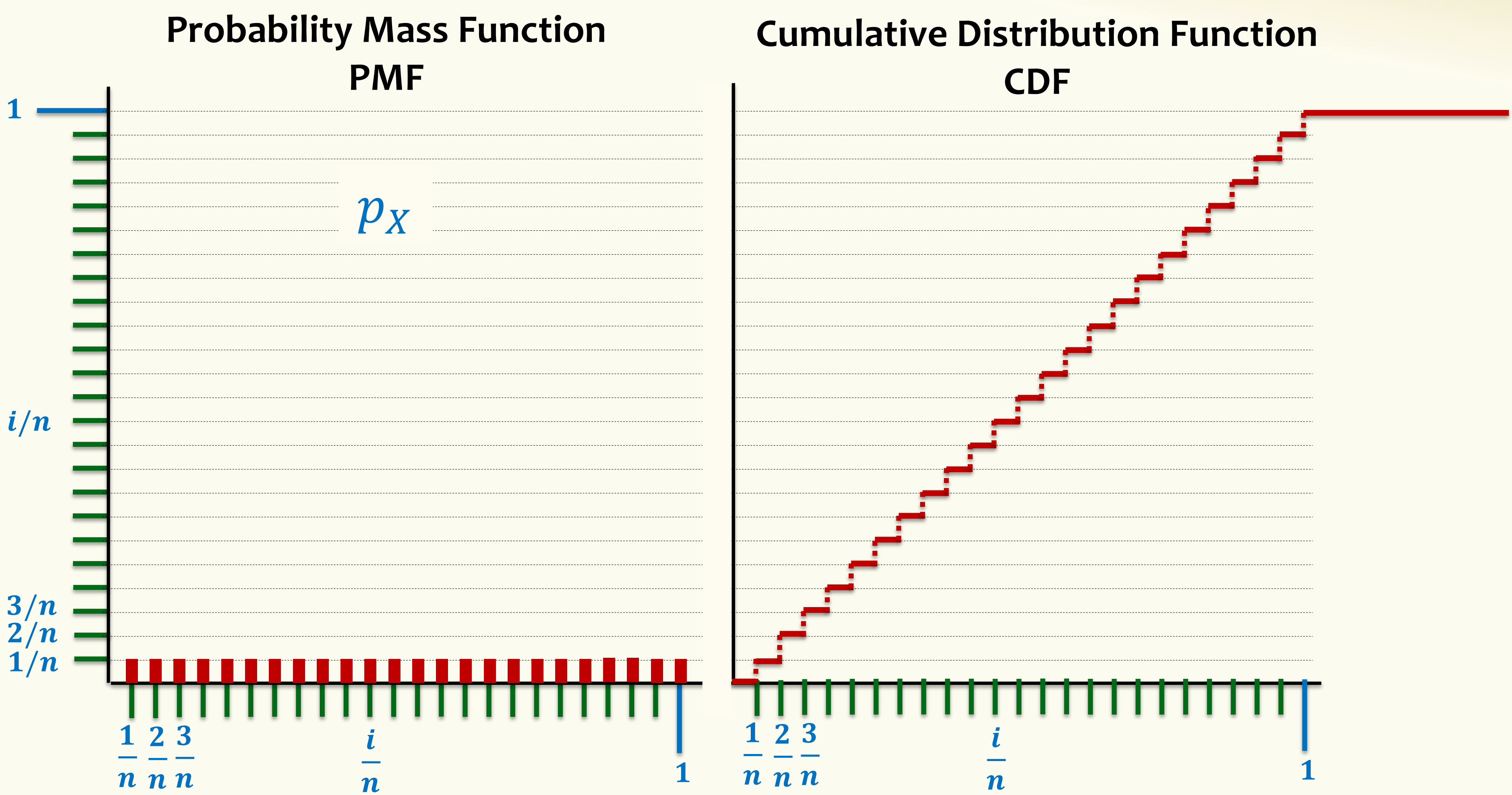


Discrete approximation?

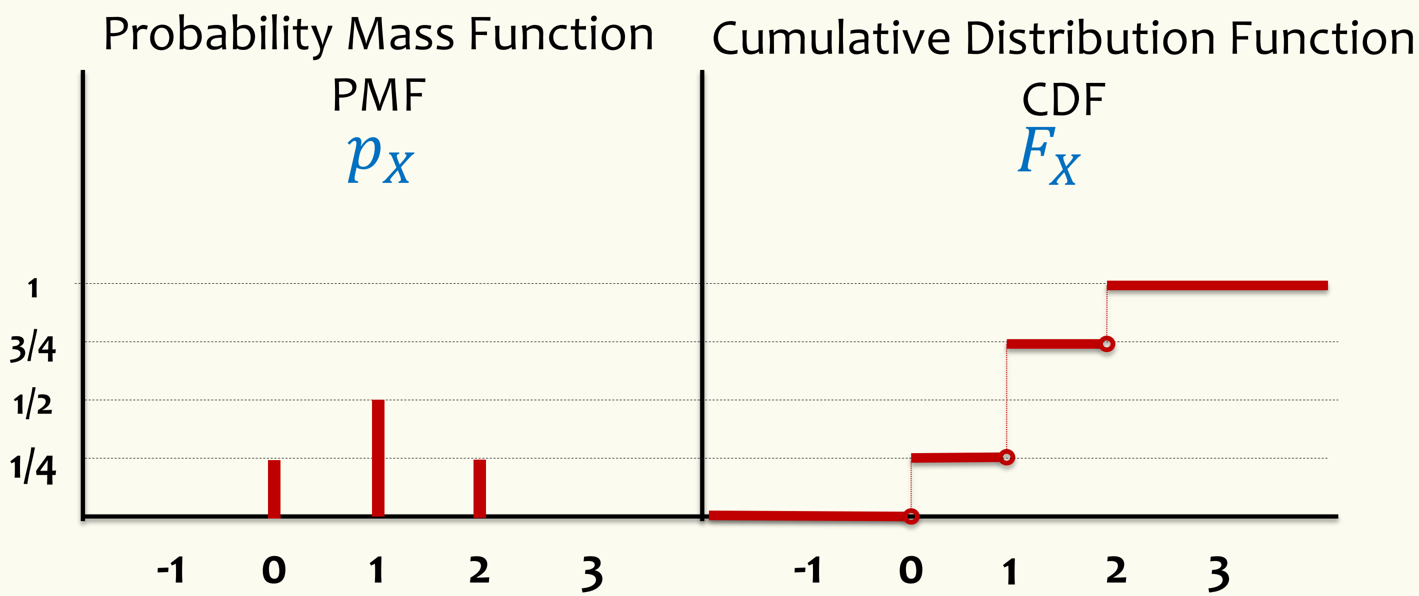
A Discrete Approximation



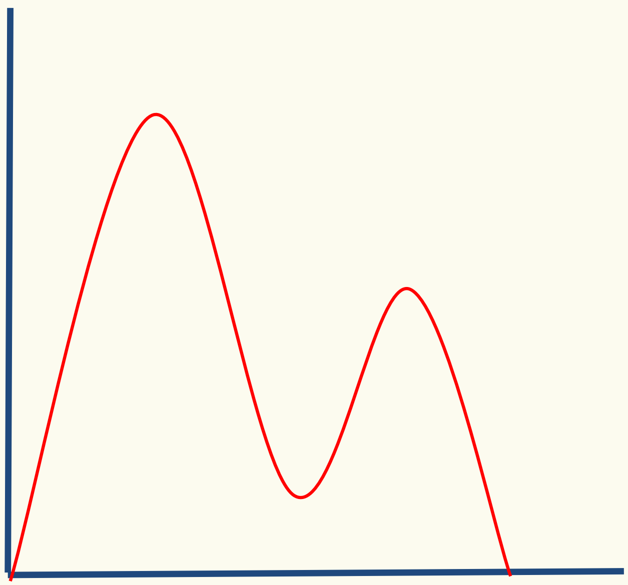
A Discrete Approximation



Recall: Cumulative Distribution Function (CDF)

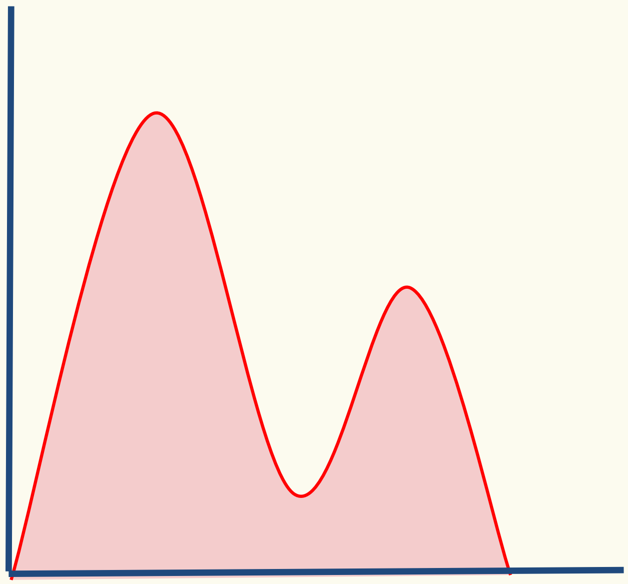


Definition. A **continuous random variable** X is defined by a **probability density function** (PDF) $f_X: \mathbb{R} \to \mathbb{R}$, such that



Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

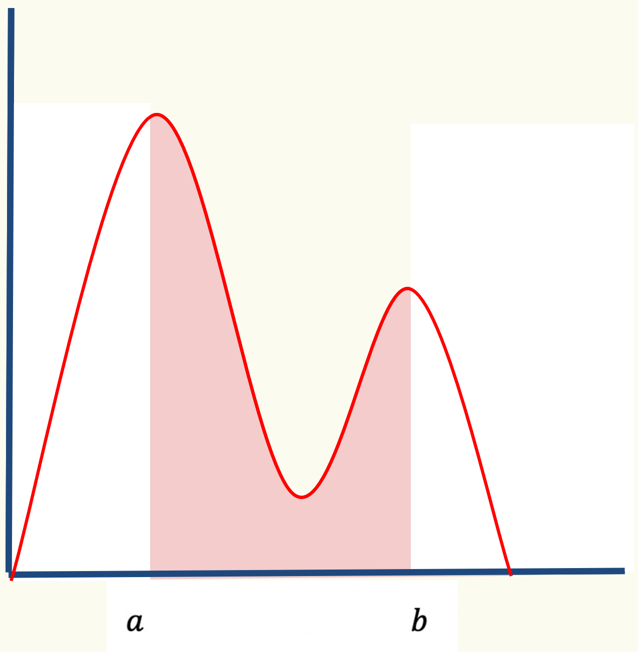
Probability Density Function - Intuition



Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\, \mathrm{d}x = 1$

Probability Density Function - Intuition

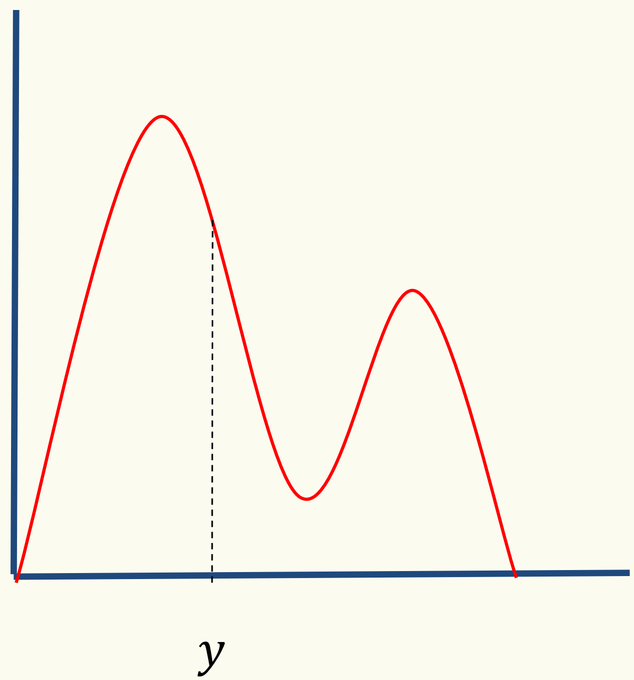


Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

$$F(b) - F(a) = P(a \leq X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x$$

Probability Density Function - Intuition



Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

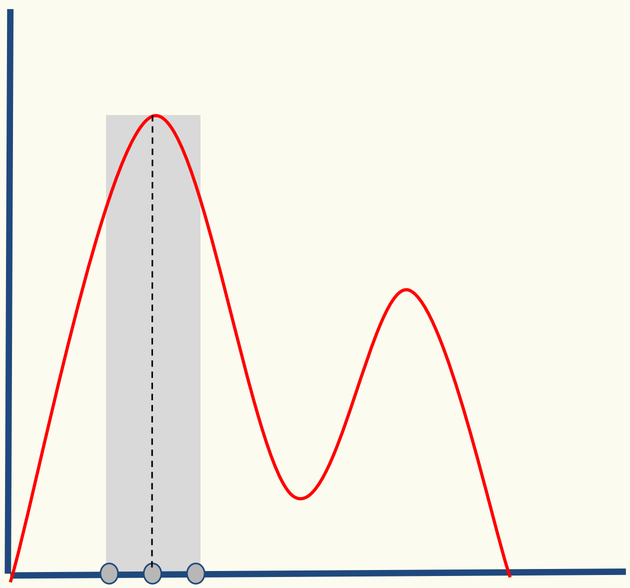
$$F(b) - F(a) = P(a \leq X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x$$

$$P(X = y) = P(y \leq X \leq y) = \int_y^y f_X(x)\,\mathrm{d}x = 0$$

Density ≠ Probability

$f_X(y) \neq 0$ $\quad$ $P(X = y) = 0$

Probability Density Function - Intuition



Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

$$F(b) - F(a) = P(a \leq X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x$$

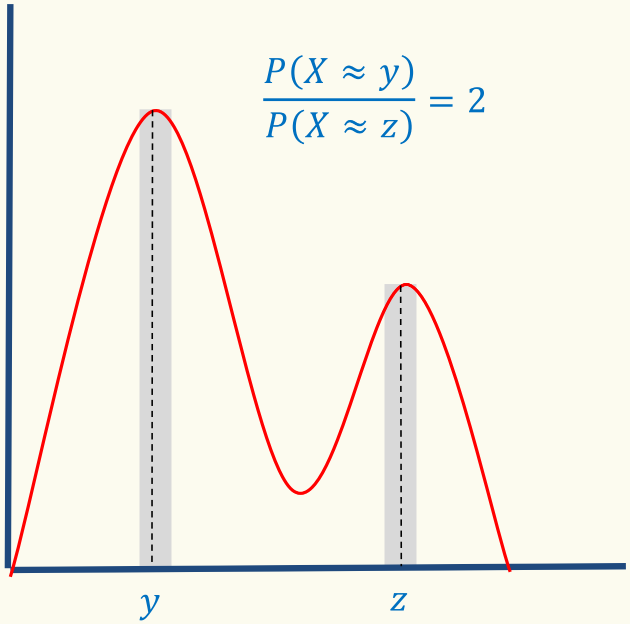
$$P(X = y) = P(y \leq X \leq y) = \int_y^y f_X(x)\,\mathrm{d}x = 0$$

$y - \frac{\epsilon}{2} \quad y \quad y + \frac{\epsilon}{2}$

$$P(X \approx y) \approx P\left(y - \frac{\epsilon}{2} \leq X \leq y + \frac{\epsilon}{2}\right) = \int_{y-\frac{\epsilon}{2}}^{y+\frac{\epsilon}{2}} f_X(x)\,\mathrm{d}x \approx \epsilon f_X(y)$$

What $f_X(x)$ measures: The local ***rate*** at which probability accumulates

Probability Density Function - Intuition



Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

$$F(b) - F(a) = P(a \leq X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x$$

$$P(X = y) = P(y \leq X \leq y) = \int_y^y f_X(x)\,\mathrm{d}x = 0$$

$$P(X \approx y) \approx P\left(y - \frac{\epsilon}{2} \leq X \leq y + \frac{\epsilon}{2}\right) = \int_{y-\frac{\epsilon}{2}}^{y+\frac{\epsilon}{2}} f_X(x)\,\mathrm{d}x \approx \epsilon f_X(y)$$

$$\frac{P(X \approx y)}{P(X \approx z)} \approx \frac{\epsilon f_X(y)}{\epsilon f_X(z)} = \frac{f_X(y)}{f_X(z)}$$

Definition. A **continuous random variable** X is defined by a **probability density function** (PDF) $f_X: \mathbb{R} \to \mathbb{R}$, such that

Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

$$F(b) - F(a) = P(a \leq X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x$$

$$P(X = y) = P(y \leq X \leq y) = \int_y^y f_X(x)\,\mathrm{d}x = 0$$

$$P(X \approx y) \approx P\left(y - \frac{\epsilon}{2} \leq X \leq y + \frac{\epsilon}{2}\right) = \int_{y-\frac{\epsilon}{2}}^{y+\frac{\epsilon}{2}} f_X(x)\,\mathrm{d}x \approx \epsilon f_X(y)$$

$$\frac{P(X \approx y)}{P(X \approx z)} \approx \frac{\epsilon f_X(y)}{\epsilon f_X(z)} = \frac{f_X(y)}{f_X(z)}$$

Cumulative Distribution Function

Definition. The **cumulative distribution function (cdf)** of X is

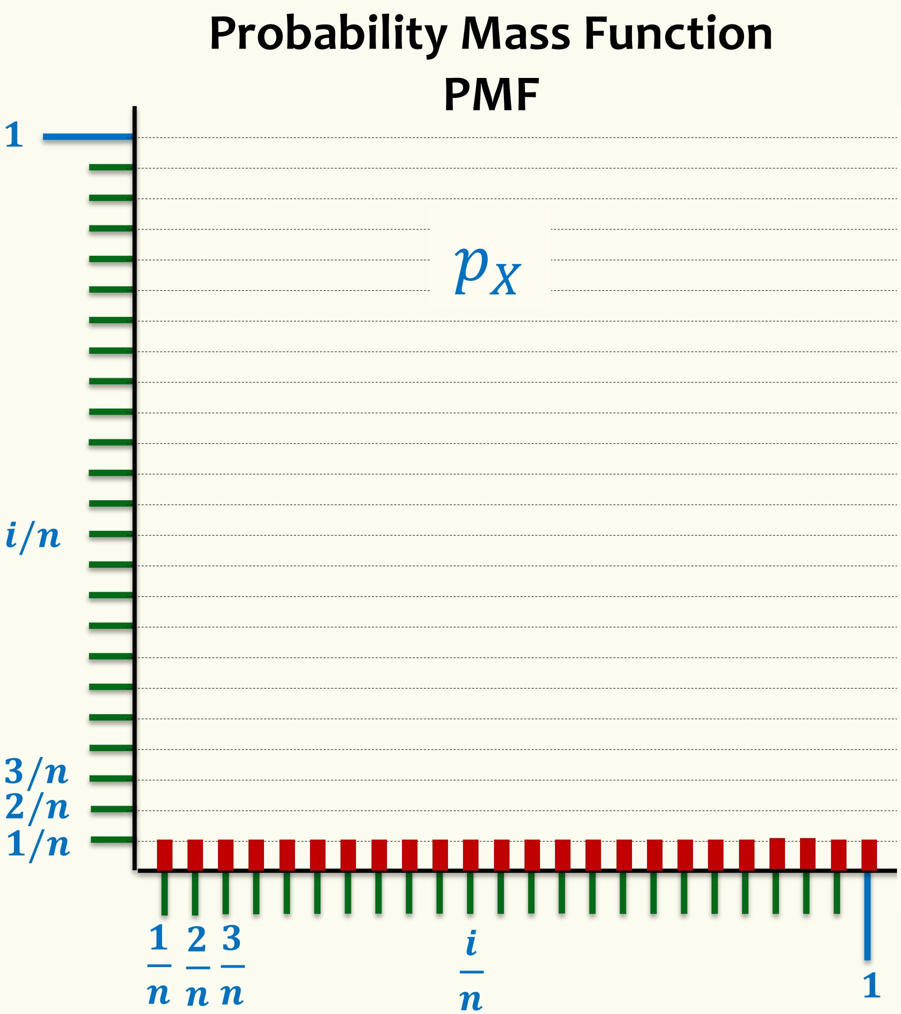
$$F_X(a) = P(X \leq a) = \int_{-\infty}^{a} f_X(x)\,\mathrm{d}x$$

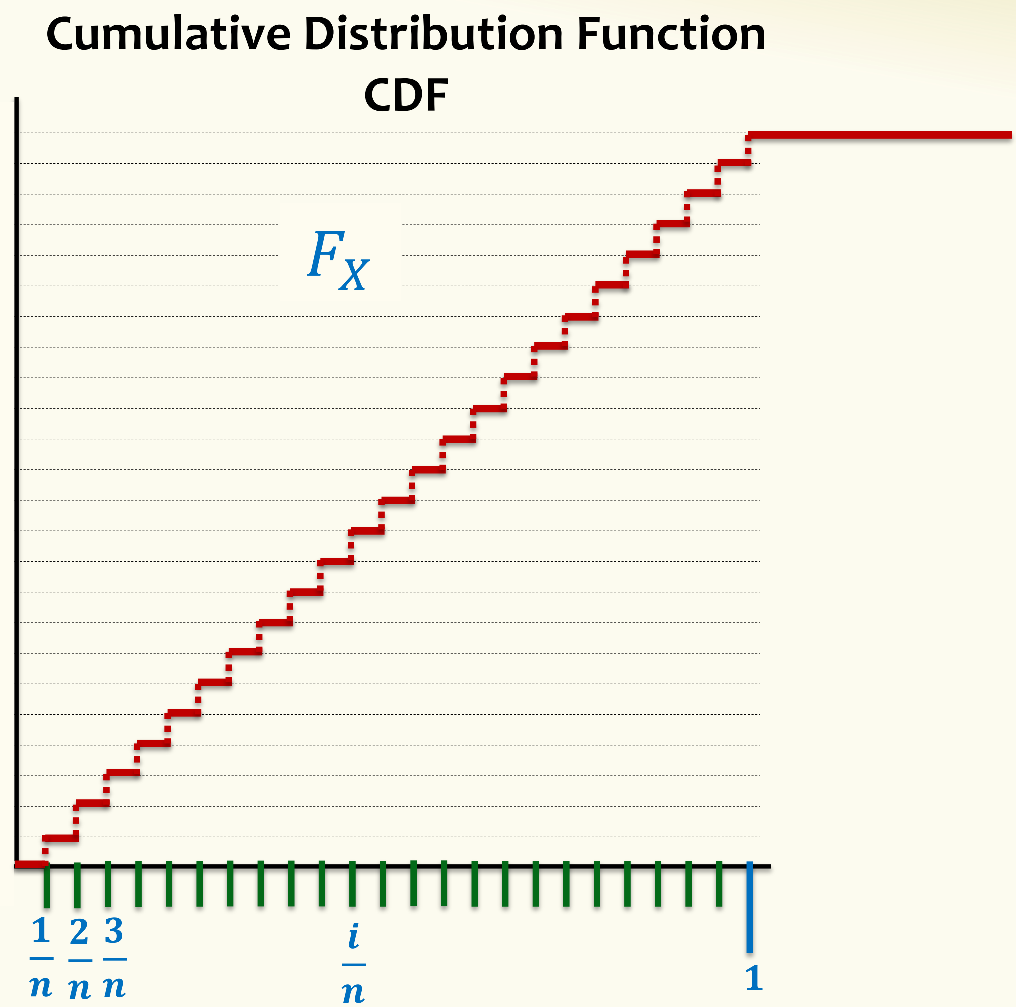
By the fundamental theorem of Calculus $f_X(x) = \frac{d}{dx} F_X(x)$

From Discrete to Continuous

	Discrete	Continuous
PMF/PDF	$p_X(x) = P(X = x)$	$f_X(x) \neq P(X = x) = 0$
CDF	$F_X(x) = \sum_{t \le x} p_X(t)$	$F_X(x) = \int_{-\infty}^{x} f_X(t)\, dt$
Normalization	$\sum_{x} p_X(x) = 1$	$\int_{-\infty}^{\infty} f_X(x)\, dx = 1$

A Discrete Approximation



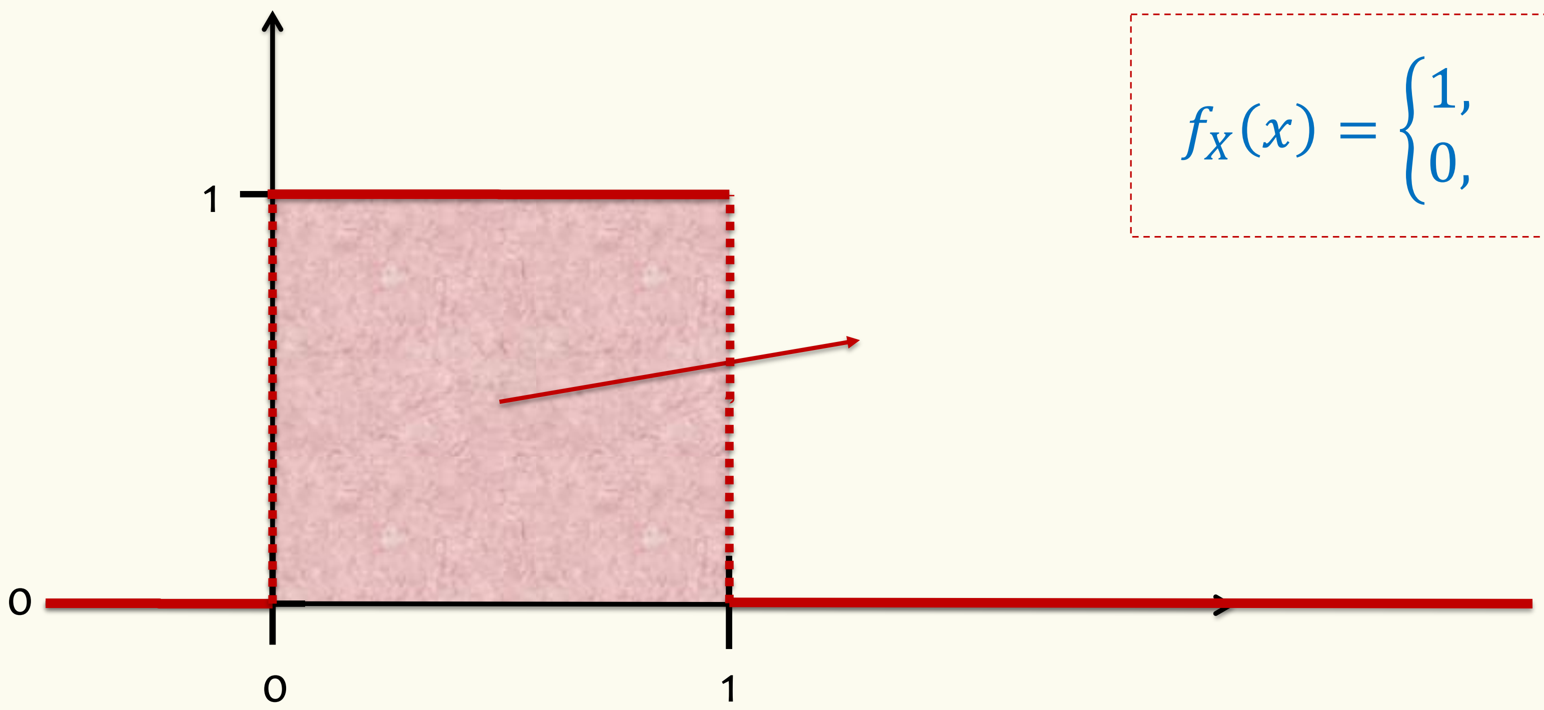


PDF of Uniform RV

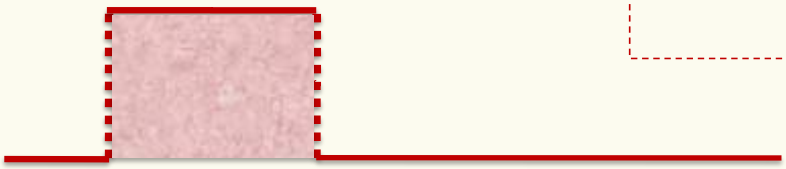
$X \sim \text{Unif}(0,1)$

$$F_X(x) = P(X \le x) = \begin{cases} 0 & x \le 0 \\ x & 0 \le x \le 1 \\ 1 & 1 \le x \end{cases}$$

$$f_X(x) = \begin{cases} 1, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}$$



$X \sim \text{Unif}(0,1)$



$$f_X(x) = \begin{cases} 1, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}$$

$$F_X(x) = P(X \le x) = \begin{cases} 0 & x \le 0 \\ x & 0 \le x \le 1 \\ 1 & 1 \le x \end{cases}$$

Non-negativity: $f_X(x) \ge 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

$$F(b) - F(a) = P(a \le X \le b) = \int_a^b f_X(x)\,\mathrm{d}x$$

$$P(X = y) = P(y \le X \le y) = \int_y^y f_X(x)\,\mathrm{d}x = 0$$

$$P(X \approx y) \approx \epsilon f_X(y)$$

$$\frac{P(X \approx y)}{P(X \approx z)} \approx \frac{\epsilon f_X(y)}{\epsilon f_X(z)} = \frac{f_X(y)}{f_X(z)}$$

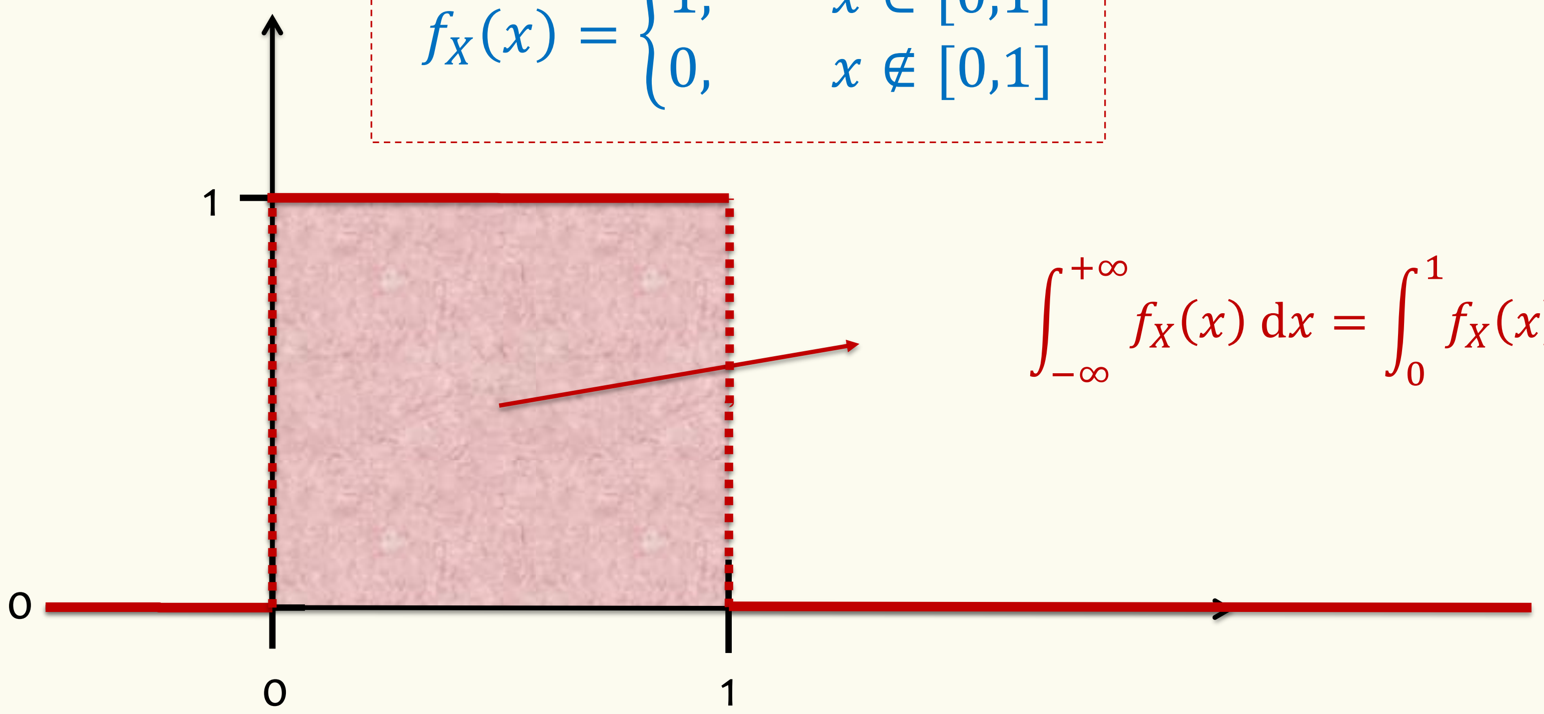
PDF of Uniform RV

$X \sim \text{Unif}(0,1)$

Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

$$f_X(x) = \begin{cases} 1, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}$$



$$\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = \int_0^1 f_X(x)\,\mathrm{d}x = 1 \cdot 1 = 1$$

Probability of Event

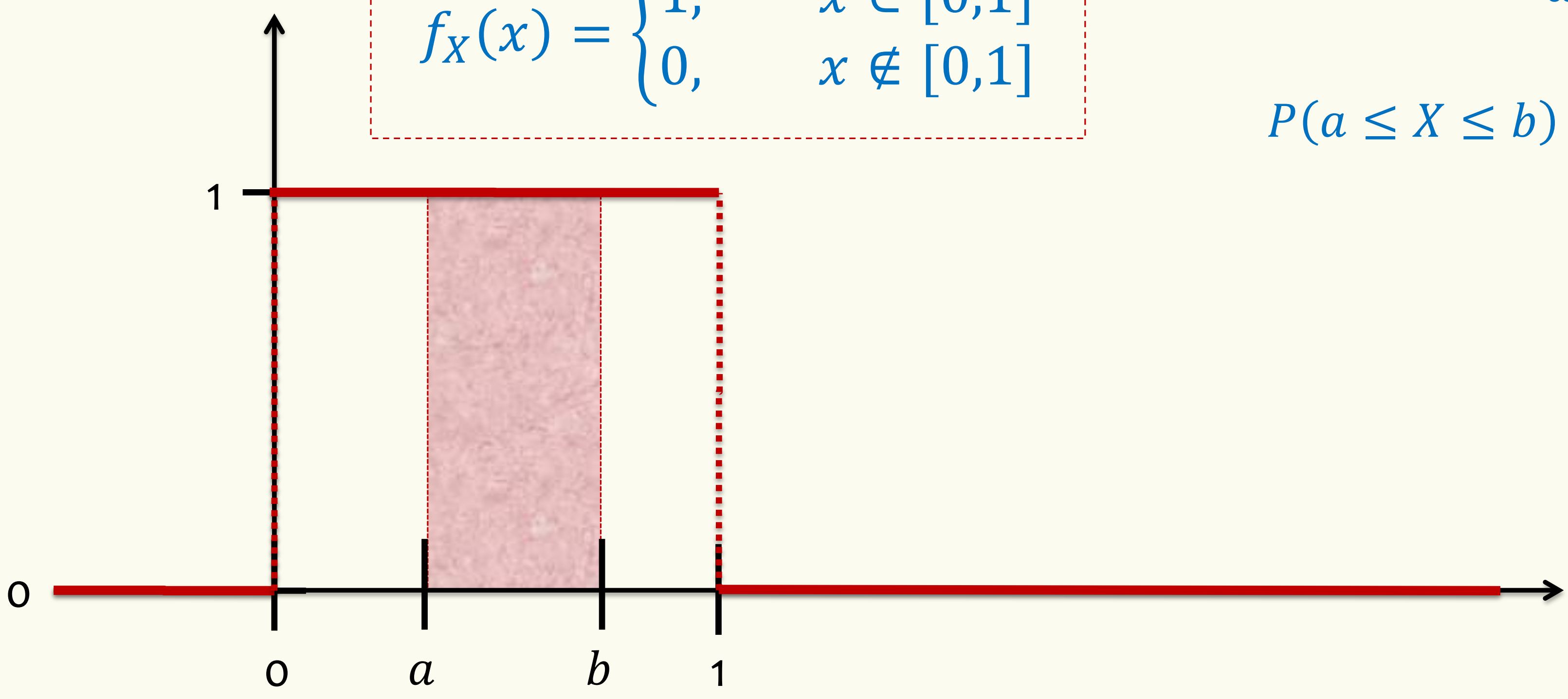
$X \sim \mathrm{Unif}(0,1)$

Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

$$f_X(x) = \begin{cases} 1, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}$$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

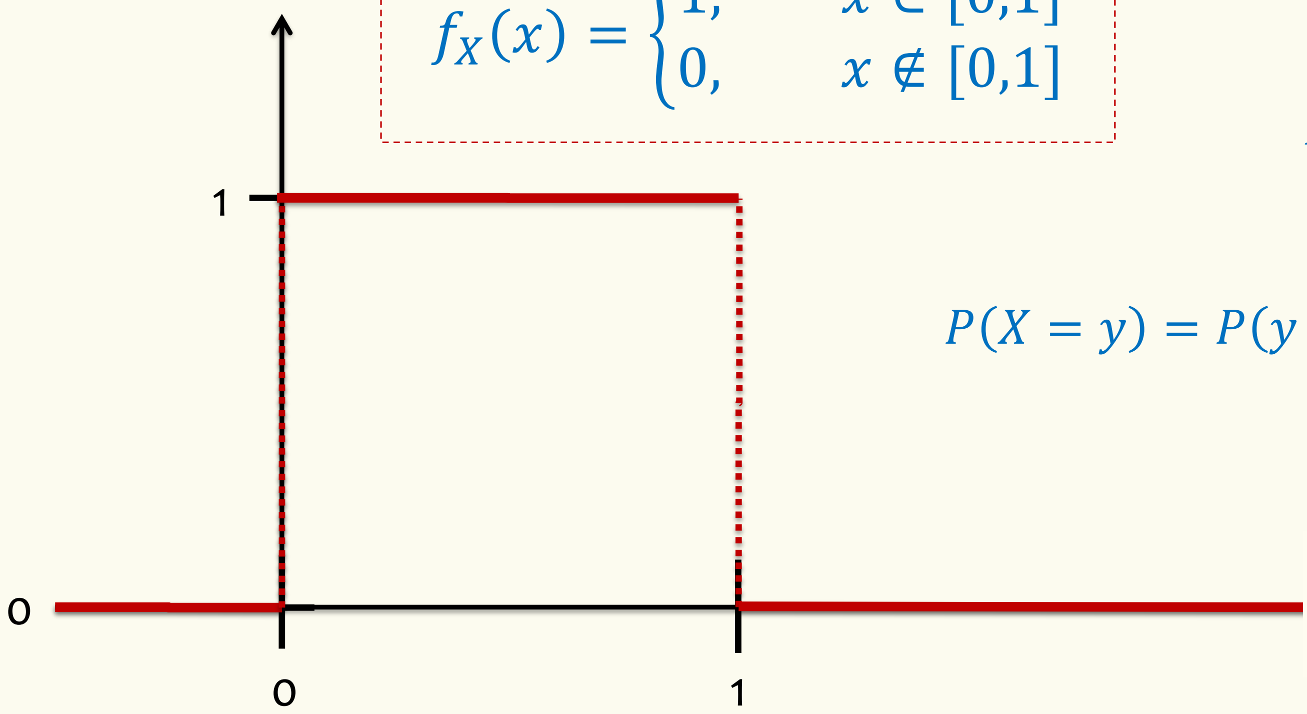
$$P(a \leq X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x$$



Probability of Event

$X \sim \text{Unif}(0,1)$

$$f_X(x) = \begin{cases} 1, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}$$



Non-negativity: $f_X(x) \geq 0$ for all $x \in \mathbb{R}$

Normalization: $\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = 1$

$$P(a \leq X \leq b) = \int_a^b f_X(x)\,\mathrm{d}x$$

$$P(X = y) = P(y \leq X \leq y) = \int_y^y f_X(x)\,\mathrm{d}x = 0$$

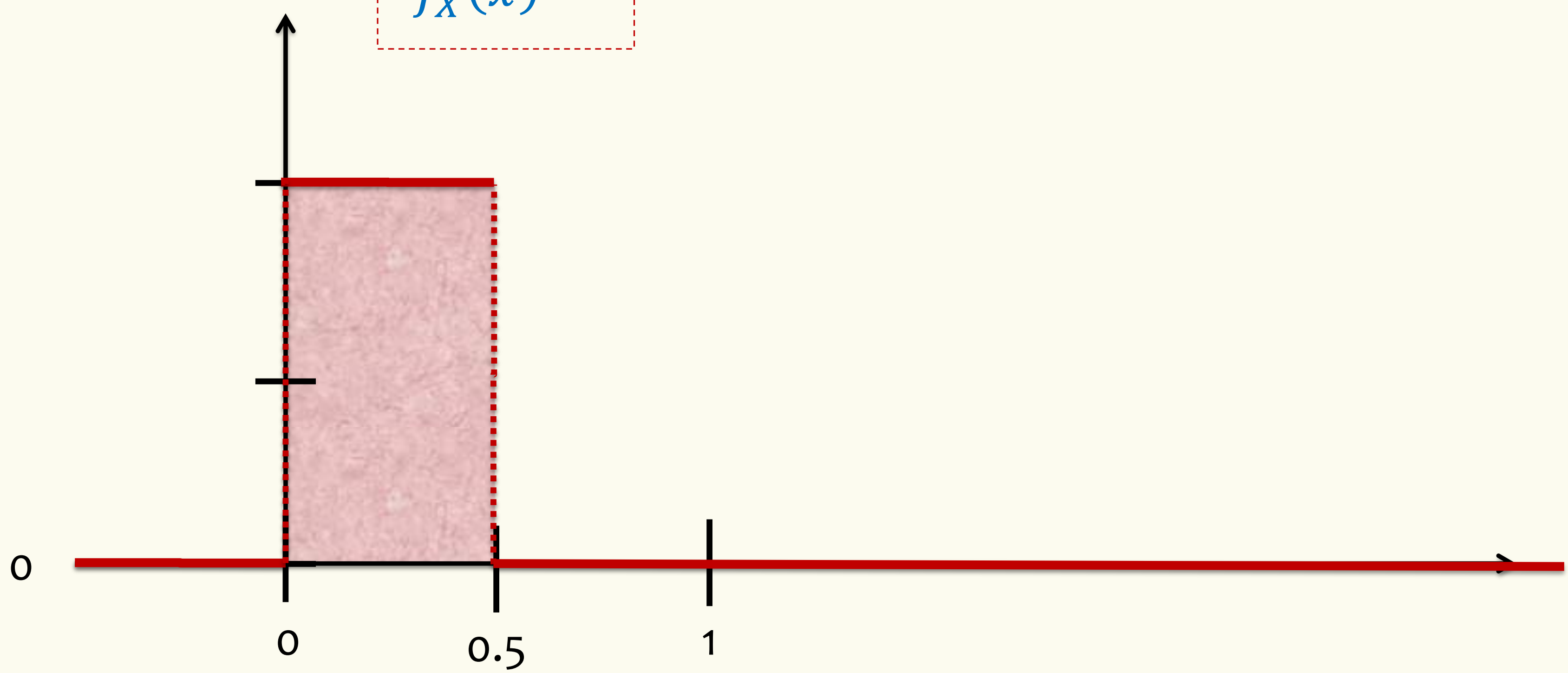
$$P(X \approx y) \approx \epsilon f_X(y) = \epsilon$$

$$\frac{P(X \approx y)}{P(X \approx z)} \approx \frac{\epsilon f_X(y)}{\epsilon f_X(z)} = \frac{f_X(y)}{f_X(z)}$$

PDF of Uniform RV

$X \sim \text{Unif}(0,0.5)$

$f_X(x) =$



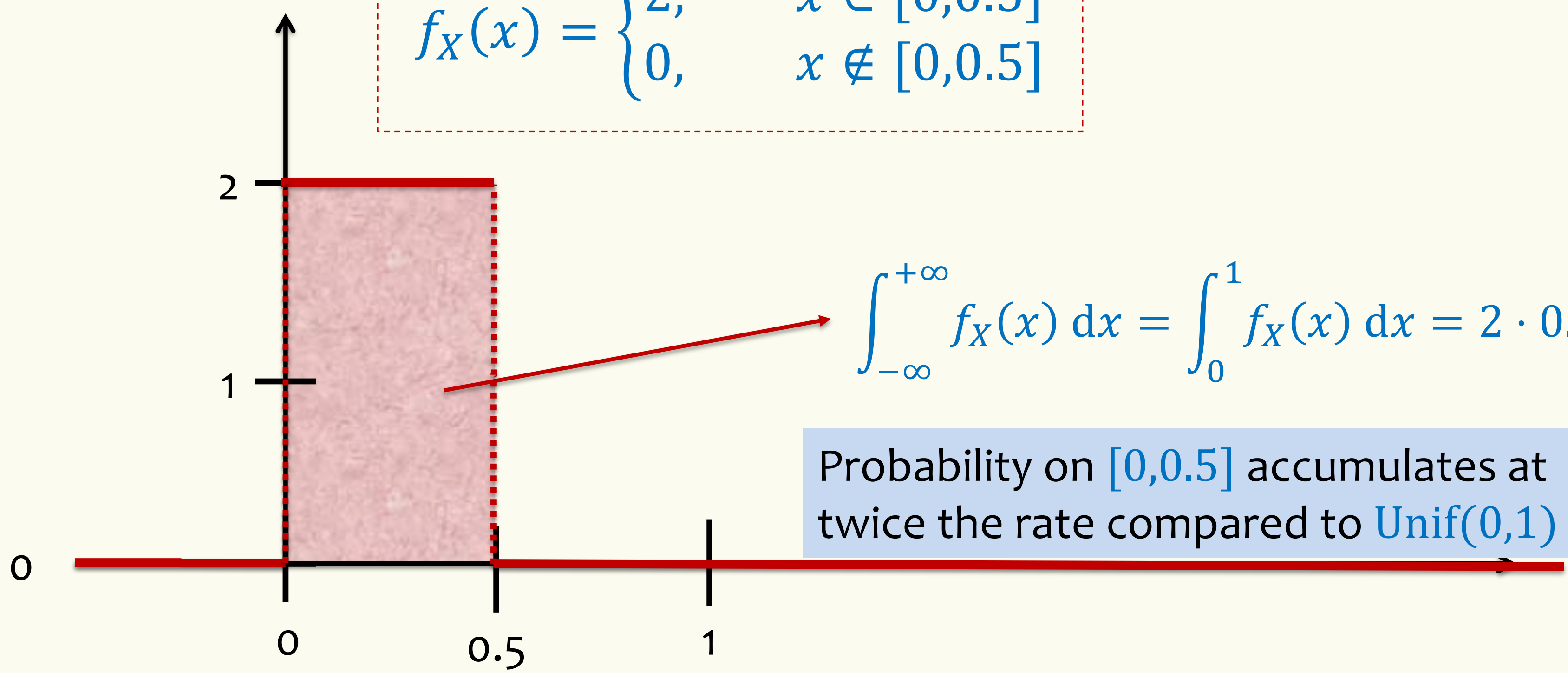
PDF of Uniform RV

$X \sim \text{Unif}(0,0.5)$

Density ≠ Probability

$f_X(x) \gg 1$ is possible!

$$f_X(x) = \begin{cases} 2, & x \in [0,0.5] \\ 0, & x \notin [0,0.5] \end{cases}$$



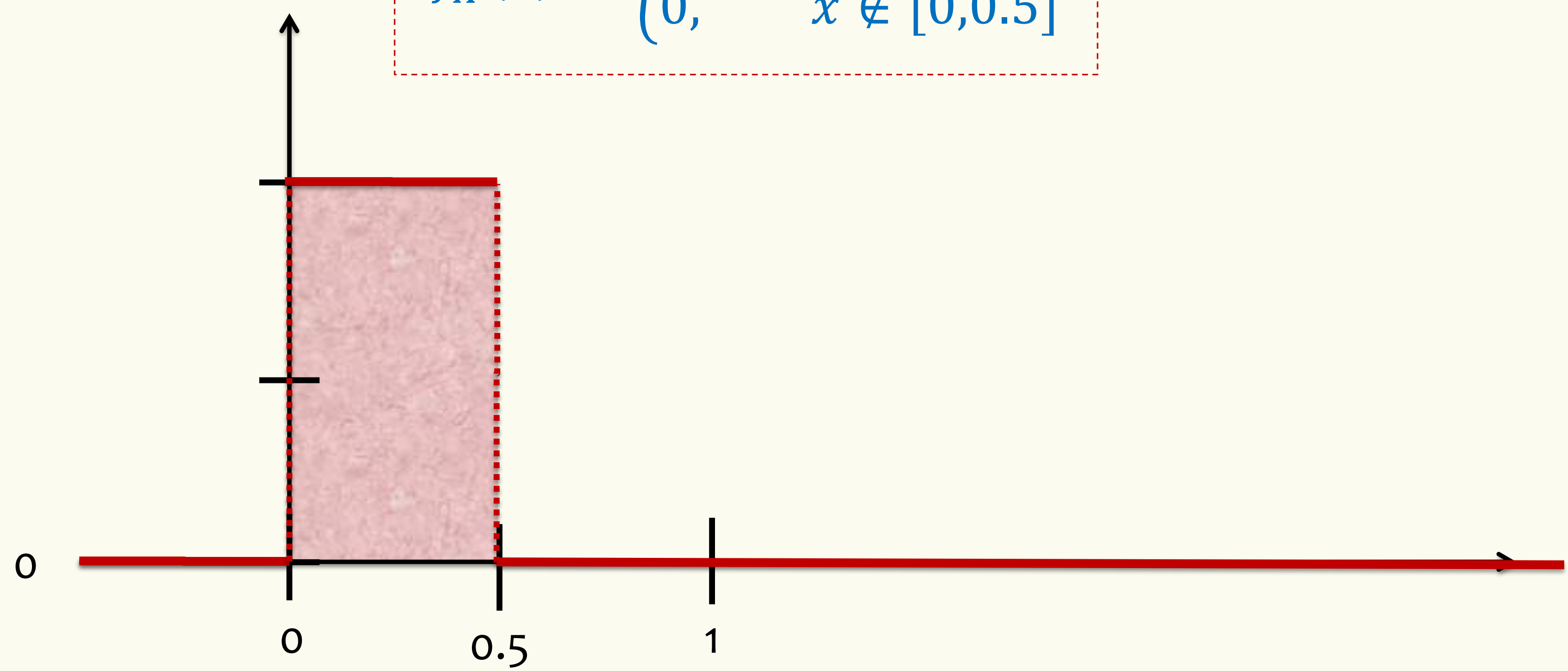
$$\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = \int_0^1 f_X(x)\,\mathrm{d}x = 2 \cdot 0.5 = 1$$

Probability on $[0,0.5]$ accumulates at twice the rate compared to $\text{Unif}(0,1)$

PDF of Uniform RV

$X \sim \text{Unif}(0,0.5)$

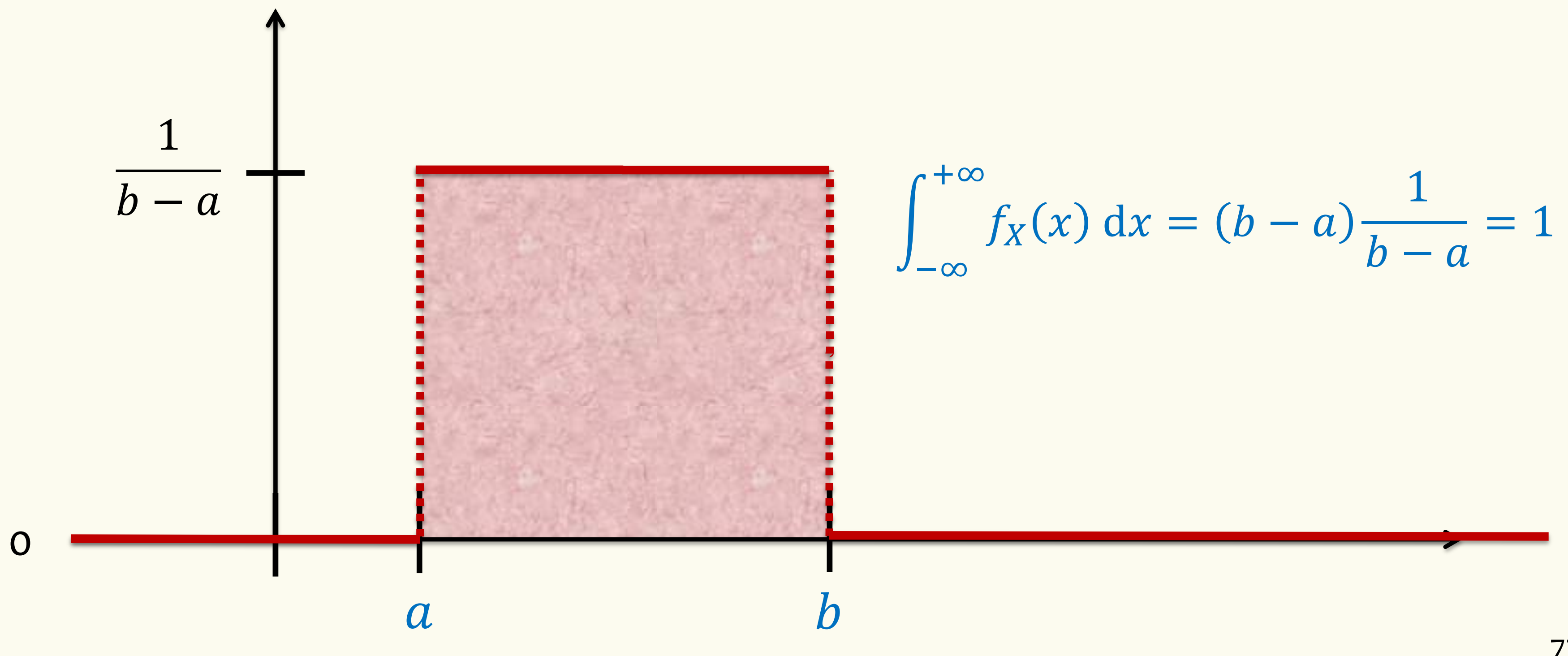
$$f_X(x) = \begin{cases} 2, & x \in [0,0.5] \\ 0, & x \notin [0,0.5] \end{cases}$$



Uniform Distribution

$X \sim \text{Unif}(a, b)$

$$f_X(x) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & \text{else} \end{cases}$$



$$\int_{-\infty}^{+\infty} f_X(x)\,\mathrm{d}x = (b-a)\frac{1}{b-a} = 1$$

Cumulative Distribution Function

Definition. The **cumulative distribution function (cdf)** of X is

$$F_X(a) = P(X \leq a) = \int_{-\infty}^{a} f_X(x)\,\mathrm{d}x$$

By the fundamental theorem of Calculus $f_X(x) = \frac{d}{dx} F_X(x)$

From Discrete to Continuous

	Discrete	Continuous
PMF/PDF	$p_X(x) = P(X = x)$	$f_X(x) \neq P(X = x) = 0$
CDF	$F_X(x) = \sum_{t \leq x} p_X(t)$	$F_X(x) = \int_{-\infty}^{x} f_X(t)\, dt$
Normalization	$\sum_{x} p_X(x) = 1$	$\int_{-\infty}^{\infty} f_X(x)\, dx = 1$

Cumulative Distribution Function

Definition. The **cumulative distribution function (cdf)** of X is

$$F_X(a) = P(X \leq a) = \int_{-\infty}^{a} f_X(x)\,\mathrm{d}x$$

By the fundamental theorem of Calculus $f_X(x) = \frac{d}{dx} F_X(x)$

Therefore: $P(X \in [a, b]) = F_X(b) - F_X(a)$

F_X is monotone increasing, since $f_X(x) \geq 0$. That is $F_X(c) \leq F_X(d)$ for $c \leq d$

$\lim_{a \to -\infty} F_X(a) = P(X \leq -\infty) = 0$ $\lim_{a \to +\infty} F_X(a) = P(X \leq +\infty) = 1$

Agenda

- Wrap-up of Poisson RVs
- Continuous Random Variables
- Probability Density Function
- Cumulative Distribution Function
- Expectation and Variance of continuous r.v.

Expectation of a Continuous RV

Definition. The **expected value** of a continuous RV X is defined as

$$\mathbb{E}[X] = \int_{-\infty}^{+\infty} f_X(x) \cdot x \, \mathrm{d}x$$

Fact. $\mathbb{E}[aX + bY + c] = a\mathbb{E}[X] + b\mathbb{E}[Y] + c$

Proof follows same ideas as discrete case

Expectation of a Continuous RV

Definition. The **expected value** of a continuous RV X is defined as

$$\mathbb{E}[X] = \int_{-\infty}^{+\infty} f_X(x) \cdot x \, \mathrm{d}x$$

Fact. $\mathbb{E}[aX + bY + c] = a\mathbb{E}[X] + b\mathbb{E}[Y] + c$

Proofs follow same ideas as discrete case

Definition. The **variance** of a continuous RV X is defined as

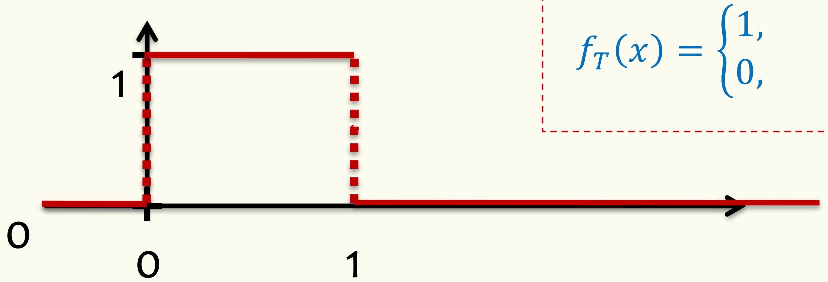
$$\mathrm{Var}(X) = \int_{-\infty}^{+\infty} f_X(x) \cdot (x - \mathbb{E}[X])^2 \, \mathrm{d}x = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

Expectation of a Continuous RV

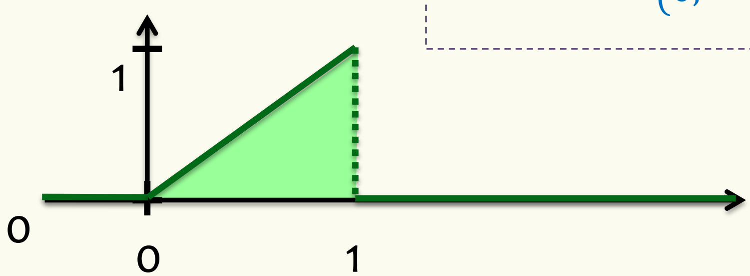
Definition.

$$\mathbb{E}[X] = \int_{-\infty}^{+\infty} f_X(x) \cdot x \, \mathrm{d}x$$

Example. $T \sim \mathrm{Unif}(0,1)$



$$f_T(x) = \begin{cases} 1, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}$$



$$f_T(x) \cdot x = \begin{cases} x, & x \in [0,1] \\ 0, & x \notin [0,1] \end{cases}$$

$$\mathbb{E}[T] = \underbrace{\frac{1}{2} 1^2}_{\text{Area of triangle}} = \frac{1}{2}$$

Area of triangle

Uniform Density – Expectation

$X \sim \text{Unif}(a, b)$

$$f_X(x) = \begin{cases} \dfrac{1}{b-a} & x \in [a, b] \\ 0 & \text{else} \end{cases}$$

$$\mathbb{E}[X] = \int_{-\infty}^{+\infty} f_X(x) \cdot x \, \mathrm{d}x$$

$$= \frac{1}{b-a} \int_a^b x \, \mathrm{d}x \quad = \frac{1}{b-a} \left(\frac{x^2}{2} \right) \Bigg|_a^b = \frac{1}{b-a} \left(\frac{b^2 - a^2}{2} \right)$$

$$= \frac{(b-a)(a+b)}{2(b-a)} \quad = \frac{a+b}{2}$$

Uniform Density – Variance

$$f_X(x) = \begin{cases} \dfrac{1}{b-a} & x \in [a,b] \\ 0 & \text{else} \end{cases}$$

$X \sim \mathrm{Unif}(a,b)$

$$\mathbb{E}[X^2] = \int_{-\infty}^{+\infty} f_X(x) \cdot x^2 \, \mathrm{d}x$$

$$= \frac{1}{b-a} \int_a^b x^2 \, \mathrm{d}x = \frac{1}{b-a} \left(\frac{x^3}{3} \right) \Bigg|_a^b = \frac{b^3 - a^3}{3(b-a)}$$

$$= \frac{(b-a)(b^2 + ab + a^2)}{3(b-a)} = \frac{b^2 + ab + a^2}{3}$$

Uniform Density – Variance

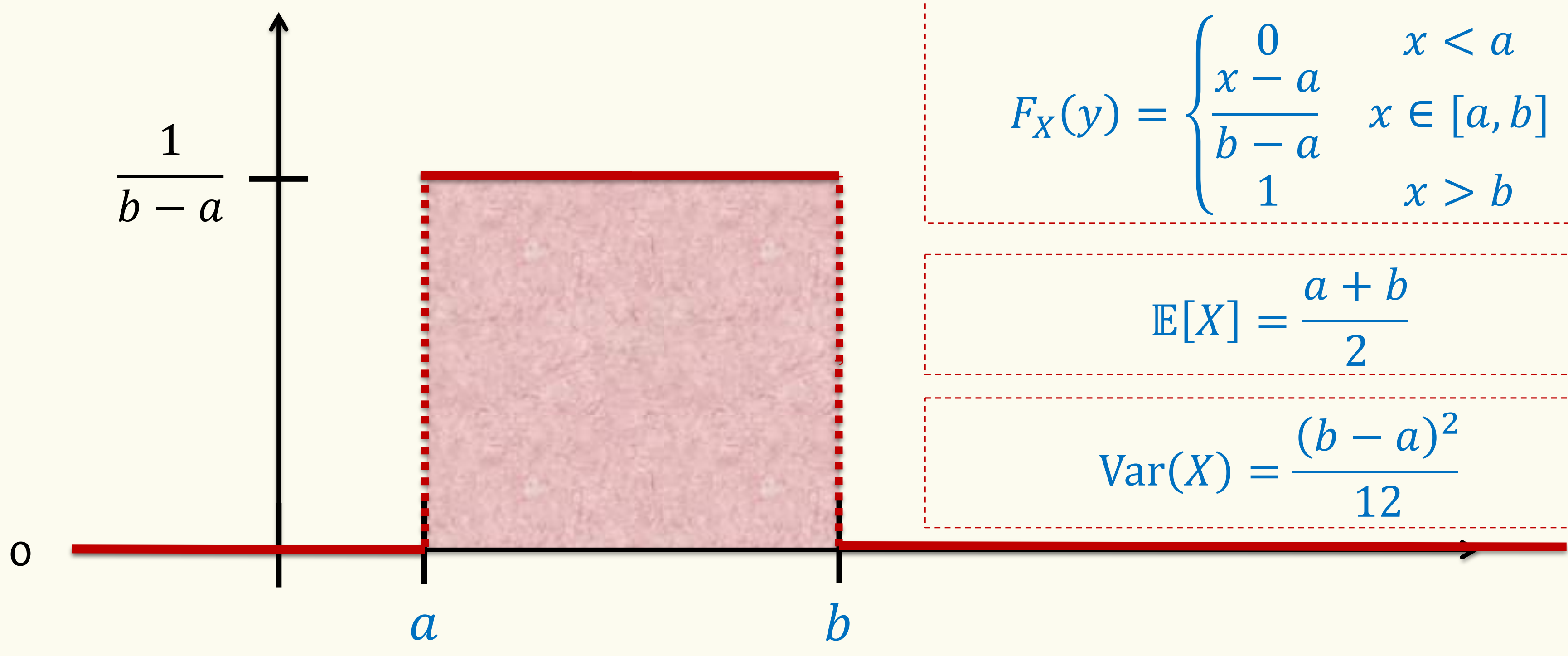
$$\mathbb{E}[X^2] = \frac{b^2 + ab + a^2}{3} \qquad \mathbb{E}[X] = \frac{a + b}{2}$$

$X \sim \text{Unif}(a, b)$

$$\begin{aligned}
\text{Var}(X) &= \mathbb{E}[X^2] - \mathbb{E}[X]^2 \\
&= \frac{b^2 + ab + a^2}{3} - \frac{a^2 + 2ab + b^2}{4} \\
&= \frac{4b^2 + 4ab + 4a^2}{12} - \frac{3a^2 + 6ab + 3b^2}{12} \\
&= \frac{b^2 - 2ab + a^2}{12} = \frac{(b - a)^2}{12}
\end{aligned}$$

Uniform Distribution Summary

$X \sim \text{Unif}(a, b)$



$$f_X(x) = \begin{cases} \frac{1}{b-a} & x \in [a,b] \\ 0 & \text{else} \end{cases}$$

$$F_X(y) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & x \in [a,b] \\ 1 & x > b \end{cases}$$

$$\mathbb{E}[X] = \frac{a+b}{2}$$

$$\text{Var}(X) = \frac{(b-a)^2}{12}$$