COMPSCI 590N
Lecture 10: Probability 2

Roy J. Adams

College of Information and Computer Sciences
University of Massachusetts Amherst

Outline

Histograms and Counting

- An important tool for exploring data is visualizing the distribution it came from.

Histograms and Counting

- An important tool for exploring data is visualizing the distribution it came from.
  - Allows us to verify and inform modeling decisions.

Histograms and Counting

- An important tool for exploring data is visualizing the distribution it came from.
  - Allows us to verify and inform modeling decisions.
  - Summary statistics (e.g. mean, standard deviation, skewness, kurtosis, etc.), while important and useful, discard information about the data.

Histograms and Counting

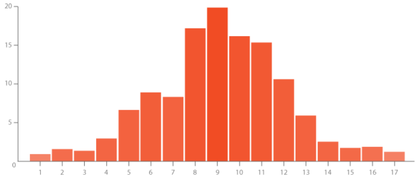
- An important tool for exploring data is visualizing the distribution it came from.
  - Allows us to verify and inform modeling decisions.
  - Summary statistics (e.g. mean, standard deviation, skewness, kurtosis, etc.), while important and useful, discard information about the data.
- A flexible tool for visualizing data distributions is the histogram.

Histograms and Counting

- An important tool for exploring data is visualizing the distribution it came from.
  - Allows us to verify and inform modeling decisions.
  - Summary statistics (e.g. mean, standard deviation, skewness, kurtosis, etc.), while important and useful, discard information about the data.
- A flexible tool for visualizing data distributions is the histogram.

Histograms and Counting

- An important tool for exploring data is visualizing the distribution it came from.
  - Allows us to verify and inform modeling decisions.
  - Summary statistics (e.g. mean, standard deviation, skewness, kurtosis, etc.), while important and useful, discard information about the data.
- A flexible tool for visualizing data distributions is the histogram.



Histograms and Counting

Numpy provides functions for counting and computing histograms.

```
>>> import numpy as np
>>> A = np.random.randn(1000)
>>> np.histogram(A,nbins=5)
>>> counts,bin_edges = np.histogram(A,bins=7)
>>> counts
array([ 38, 180, 381, 289,  98,  11,   3])
>>> bin_edges
array([-2.75925765, -1.77917762, -0.7990976 ,
    0.18098242,  1.16106244,
        2.14114246,  3.12122248,  4.1013025 ])
```

Histograms in Python

- There are a variety of tools for plotting distributions:

Histograms in Python

- There are a variety of tools for plotting distributions:
  - `matplotlib.pyplot.histogram` computes and plots a histogram from an array.

Histograms in Python

- There are a variety of tools for plotting distributions:
  - `matplotlib.pyplot.histogram` computes and plots a histogram from an array.
  - `pandas.DataFrame.hist` plots a histogram directly from a `pandas.DataFrame`.

Histograms in Python

- There are a variety of tools for plotting distributions:
  - `matplotlib.pyplot.histogram` computes and plots a histogram from an array.
  - `pandas.DataFrame.hist` plots a histogram directly from a `pandas.DataFrame`.
  - `Seaborn` is an extension to `matplotlib` design explicitly for plotting distributions.

Outline

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).
- Randomized communications (e.g. LAN).

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).
- Randomized communications (e.g. LAN).
- Approximate integration (we will look at this later).

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).
- Randomized communications (e.g. LAN).
- Approximate integration (we will look at this later).
- Randomized experiments.

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).
- Randomized communications (e.g. LAN).
- Approximate integration (we will look at this later).
- Randomized experiments.
- Statistical modeling:

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).
- Randomized communications (e.g. LAN).
- Approximate integration (we will look at this later).
- Randomized experiments.
- Statistical modeling:
    - Evaluating models.

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).
- Randomized communications (e.g. LAN).
- Approximate integration (we will look at this later).
- Randomized experiments.
- Statistical modeling:
  - Evaluating models.
  - Approximate inference.

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).
- Randomized communications (e.g. LAN).
- Approximate integration (we will look at this later).
- Randomized experiments.
- Statistical modeling:
  - Evaluating models.
  - Approximate inference.
  - Approximate estimation.

Applications of Randomness

Randomness is a core operation for many commonly used algorithms:

- Cryptography and encryption.
- Randomized storage (e.g. random hash tables used in Python dictionaries).
- Randomized communications (e.g. LAN).
- Approximate integration (we will look at this later).
- Randomized experiments.
- Statistical modeling:
  - Evaluating models.
  - Approximate inference.
  - Approximate estimation.
- **Many** other randomized algorithms.

Random Sampling

- Problem: Given a distribution $p(x)$ we would like to generate samples from this distribution.

Random Sampling

- Problem: Given a distribution $p(x)$ we would like to generate samples from this distribution.
- Example: To generate samples from a Bernoulli distribution (binary), flip a bunch of coins. The result of each coin flip is a sample from the Bernoulli.

Types of Sampling

As you might imagine, sampling has been the focus of intense study for a long time. Some common sampling algorithms are:

Types of Sampling

As you might imagine, sampling has been the focus of intense study for a long time. Some common sampling algorithms are:

- Inverse CDF sampling (we will look at this algorithm).

Types of Sampling

As you might imagine, sampling has been the focus of intense study for a long time. Some common sampling algorithms are:

- Inverse CDF sampling (we will look at this algorithm).
- Rejection sampling.

Types of Sampling

As you might imagine, sampling has been the focus of intense study for a long time. Some common sampling algorithms are:

- Inverse CDF sampling (we will look at this algorithm).
- Rejection sampling.
- Markov Chain Monte Carlo (MCMC).

Types of Sampling

As you might imagine, sampling has been the focus of intense study for a long time. Some common sampling algorithms are:

- Inverse CDF sampling (we will look at this algorithm).
- Rejection sampling.
- Markov Chain Monte Carlo (MCMC).
- Metropolis-Hastings.

Types of Sampling

As you might imagine, sampling has been the focus of intense study for a long time. Some common sampling algorithms are:

- Inverse CDF sampling (we will look at this algorithm).
- Rejection sampling.
- Markov Chain Monte Carlo (MCMC).
- Metropolis-Hastings.
- Slice sampling.

Types of Sampling

As you might imagine, sampling has been the focus of intense study for a long time. Some common sampling algorithms are:

- Inverse CDF sampling (we will look at this algorithm).
- Rejection sampling.
- Markov Chain Monte Carlo (MCMC).
- Metropolis-Hastings.
- Slice sampling.
- The list goes on.

Cumulative Distribution Functions

Given a distribution $p(x)$ for a random variable X, the associated Cumulative Distribution Function (CDF) $F(x)$, gives the probability that X is less then some value, i.e. $F(x) = P(X \leq x)$.

Cumulative Distribution Functions

Given a distribution $p(x)$ for a random variable X, the associated Cumulative Distribution Function (CDF) $F(x)$, gives the probability that X is less then some value, i.e. $F(x) = P(X \leq x)$.

Cumulative Distribution Function

The CDF $F(x)$ for a distribution $p(x)$ is given by,

$$F(x) = \int_{-\infty}^{x} p(x')dx'$$

for continuous variables and by

$$F(x) = \sum_{x'=-\infty}^{x} P(x')$$

for discrete variables.

Cumulative Distribution Functions: Multinomial

For example: Consider the CDF for $\textit{Multinomial}(1, \frac{1}{6}, ..., \frac{1}{6})$ (i.e. a six sided dice roll).

Cumulative Distribution Functions: Multinomial

For example: Consider the CDF for $Multinomial(1, \frac{1}{6}, ..., \frac{1}{6})$ (i.e. a six sided dice roll).

$$F(x) = P(X \leq x) = \sum_{i=1}^{x} \frac{1}{6}$$
$$F(3) = P(X \leq 3) = \frac{3}{6} = \frac{1}{2}$$

Cumulative Distribution Functions: Multinomial

For example: Consider the CDF for $Multinomial(1, \frac{1}{6}, ..., \frac{1}{6})$ (i.e. a six sided dice roll).

$$F(x) = P(X \leq x) = \sum_{i=1}^{x} \frac{1}{6}$$

$$F(3) = P(X \leq 3) = \frac{3}{6} = \frac{1}{2}$$

Sampling Algorithms: Inverse CDF Sampling

One of the most fundamental sampling algorithms is inverse CDF sampling. For any invertible CDF, we can use the following algorithm.

Sampling Algorithms: Inverse CDF Sampling

One of the most fundamental sampling algorithms is inverse CDF sampling. For any invertible CDF, we can use the following algorithm.

Inverse CDF Sampling

Let $F(x)$ be an invertible CDF. Then we can sample from the associated distribution by:

1. Sample $u^{(s)}$ from a Uniform(0,1).
2. $x^{(s)} = F^{-1}(U)$

Monte Carlo Integration

- **Monte Carlo Integration** is a method for approximating the value of an integral that is difficult (or impossible) to compute.

Monte Carlo Integration

- **Monte Carlo Integration** is a method for approximating the value of an integral that is difficult (or impossible) to compute.
- The core idea is based on approximating **expected values**.

Expected Values

A common computation when working with probability distributions is an expected value.

Expected Values

A common computation when working with probability distributions is an expected value.

Expected Value

Given a distribution $p(x)$ over a random variable X and a function of the random variable $f : \Omega \to \mathbb{R}$, the expected valued of the function is defined as

$$\mathbb{E}[f(X)] = \int_{\Omega} p(x)f(x)dx$$

where the integral is a sum when X is discrete.

Expected Values: Multinomial

- Let X be the result of a fair six-sided dice roll.

Expected Values: Multinomial

- Let X be the result of a fair six-sided dice roll.
- What is $\mathbb{E}[X]$?

Expected Values: Multinomial

- Let X be the result of a fair six-sided dice roll.
- What is $\mathbb{E}[X]$?

Expected Values: Multinomial

- Let X be the result of a fair six-sided dice roll.
- What is $\mathbb{E}[X]$?

$$\begin{aligned}\mathbb{E}[X] &= \sum_x P(X = x) \cdot X \\ &= \sum_{x=1}^{6} \frac{1}{6} \cdot x \\ &= \frac{1}{6}(1 + 2 + 3 + 4 + 5 + 6) = 3.5\end{aligned}$$

Expected Values: Normal

- Let X be a Normally distributed random variable with mean μ and standard deviation σ.

Expected Values: Normal

- Let X be a Normally distributed random variable with mean μ and standard deviation σ.
- What is $\mathbb{E}[X^2]$?

Expected Values: Normal

- Let X be a Normally distributed random variable with mean μ and standard deviation σ.
- What is $\mathbb{E}[X^2]$?

Expected Values: Normal

- Let X be a Normally distributed random variable with mean μ and standard deviation σ.
- What is $\mathbb{E}[X^2]$?

$$\begin{aligned}\mathbb{E}[X^2] &= \int_x \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) x^2 dx \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_x \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) x^2 dx \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_x \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) x^2 dx\end{aligned}$$

The Importance of Expected Values

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.
- This may occur because:

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.
- This may occur because:
  - The normalization constant of the distribution is difficult to evaluate.

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.
- This may occur because:
    - The normalization constant of the distribution is difficult to evaluate.
    - There is not analytical solution to the integral.

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.
- This may occur because:
  - The normalization constant of the distribution is difficult to evaluate.
  - There is not analytical solution to the integral.
- What do we do? Approximate!

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.
- This may occur because:
  - The normalization constant of the distribution is difficult to evaluate.
  - There is not analytical solution to the integral.
- What do we do? Approximate!
- Types of approximations:

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.
- This may occur because:
  - The normalization constant of the distribution is difficult to evaluate.
  - There is not analytical solution to the integral.
- What do we do? Approximate!
- Types of approximations:
  1. Approximate the distribution with a simpler one that allows us to take the expectation (e.g. Variational Bayes).

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.
- This may occur because:
  - The normalization constant of the distribution is difficult to evaluate.
  - There is not analytical solution to the integral.
- What do we do? Approximate!
- Types of approximations:
  1. Approximate the distribution with a simpler one that allows us to take the expectation (e.g. Variational Bayes).
  2. Upper and lower-bound based approximations (e.g. Variational Approximations).

Difficult Expected Values

- In many cases, the expected value is too complex to evaluate.
- This may occur because:
  - The normalization constant of the distribution is difficult to evaluate.
  - There is not analytical solution to the integral.
- What do we do? Approximate!
- Types of approximations:
  1. Approximate the distribution with a simpler one that allows us to take the expectation (e.g. Variational Bayes).
  2. Upper and lower-bound based approximations (e.g. Variational Approximations).
  3. Sampling based approximations (Monte Carlo Integration).

Monte Carlo Integration

Monte Carlo integration approximates an expected value by sampling from the associated distribution and then taking an average.

Monte Carlo Integration

Monte Carlo integration approximates an expected value by sampling from the associated distribution and then taking an average.

Monte Carlo Integration

Let X be a random variable drawn from $p(x)$, let $f(x)$ be a function of X, and let $x^{(1)}, \ldots, x^{(S)}$ be samples from $p(x)$, then

$$\mathbb{E}(f(x)) \approx \frac{1}{S} \sum_{s} x^{(s)}$$

Monte Carlo Integration

How can we use Monte Carlo integration to calculate general integrals?

Monte Carlo Integration

How can we use Monte Carlo integration to calculate general integrals?

$$\int_a^b f(x)dx = \int_a^b \frac{b-a}{b-a} f(x)dx$$
$$= (b-a)\int_a^b \frac{1}{b-a} f(x)dx$$
$$= (b-a)\mathbb{E}[f(X)]$$

where X is draw from $Uniform(a, b)$.

Monte Carlo Integration

How can we use Monte Carlo integration to calculate general integrals?

$$\begin{aligned}\int_a^b f(x)dx &= \int_a^b \frac{b-a}{b-a} f(x)dx \\ &= (b-a)\int_a^b \frac{1}{b-a} f(x)dx \\ &= (b-a)\mathbb{E}[f(X)]\end{aligned}$$

where X is draw from $Uniform(a, b)$.

- Now we can approximate this expected value using sampling.

Monte Carlo Integration

How can we use Monte Carlo integration to calculate general integrals?

$$\int_a^b f(x)dx = \int_a^b \frac{b-a}{b-a} f(x)dx$$
$$= (b-a)\int_a^b \frac{1}{b-a} f(x)dx$$
$$= (b-a)\mathbb{E}[f(X)]$$

where X is draw from $Uniform(a,b)$.

- Now we can approximate this expected value using sampling.
- This works for multidimensional integrals as well.

Monte Carlo Integration: Area of a Circle

- How can we use Monte Carlo integration to estimate the area of a circle with radius r?

Monte Carlo Integration: Area of a Circle

- How can we use Monte Carlo integration to estimate the area of a circle with radius r?
- Step 1: Formulate the problem as a bounded integral.

Monte Carlo Integration: Area of a Circle

- How can we use Monte Carlo integration to estimate the area of a circle with radius r?
- Step 1: Formulate the problem as a bounded integral.

Monte Carlo Integration: Area of a Circle

- How can we use Monte Carlo integration to estimate the area of a circle with radius r?
- Step 1: Formulate the problem as a bounded integral.

A circle is defined by $x^2 + y^2 = r^2$ so we can define a piecewise function that tells us whether we are inside or outside of the circle.

$$f(x,y) = \begin{cases} 1 & : x^2 + y^2 \leq r^2 \\ 0 & : x^2 + y^2 > r^2 \end{cases}$$

Monte Carlo Integration: Area of a Circle

- How can we use Monte Carlo integration to estimate the area of a circle with radius r?
- Step 1: Formulate the problem as a bounded integral.

A circle is defined by $x^2 + y^2 = r^2$ so we can define a piecewise function that tells us whether we are inside or outside of the circle.

$$f(x,y) = \begin{cases} 1 & : x^2 + y^2 \leq r^2 \\ 0 & : x^2 + y^2 > r^2 \end{cases}$$

The area of a circle is given by

$$A_r = \int_{-r}^{r} \int_{-r}^{r} f(x,y) dx dy$$

Monte Carlo Integration: Area of a Circle

- Now that the area is formulated as an integral, how do we approximate it with sampling?

Monte Carlo Integration: Area of a Circle

- Now that the area is formulated as an integral, how do we approximate it with sampling?
- Step 1: Identify the sampling distribution.

Monte Carlo Integration: Area of a Circle

- Now that the area is formulated as an integral, how do we approximate it with sampling?
- Step 1: Identify the sampling distribution.

Monte Carlo Integration: Area of a Circle

- Now that the area is formulated as an integral, how do we approximate it with sampling?
- Step 1: Identify the sampling distribution.
- In this case we are integrating over the $2r \times 2r$ square.

Monte Carlo Integration: Area of a Circle

- Now that the area is formulated as an integral, how do we approximate it with sampling?
- Step 1: Identify the sampling distribution.

- In this case we are integrating over the $2r \times 2r$ square.
- How do we sample uniformly from a square?

Monte Carlo Integration: Area of a Circle

- Now that the area is formulated as an integral, how do we approximate it with sampling?
- Step 1: Identify the sampling distribution.

- In this case we are integrating over the $2r \times 2r$ square.
- How do we sample uniformly from a square?
- Sample x and y independently from $Uniform(-r, r)$

Monte Carlo Integration: Area of a Circle

Now we can estimate the area of a circle as:

$$A_r = \int_{-r}^{r}\int_{-r}^{r} f(x,y)dxdy$$
$$\approx 2r^2 \frac{1}{S}\sum_{s} f(x^{(s)}, y^{(s)})$$

where

$$f(x,y) == \begin{cases} 1 & : x^2 + y^2 <= r^2 \\ 0 & : x^2 + y^2 > r^2 \end{cases}$$

Monte Carlo Integration: Area of a Circle

Outline

Sources of Randomness

- Most sampling methods are based on the assumption that we can generate a uniform random number.

Sources of Randomness

- Most sampling methods are based on the assumption that we can generate a uniform random number.
- How do we generate a uniform random number?

Sources of Randomness

- Most sampling methods are based on the assumption that we can generate a uniform random number.
- How do we generate a uniform random number?
- There are two main methods:

Sources of Randomness

- Most sampling methods are based on the assumption that we can generate a uniform random number.
- How do we generate a uniform random number?
- There are two main methods:
  1. Harvested natural randomness

Sources of Randomness

- Most sampling methods are based on the assumption that we can generate a uniform random number.
- How do we generate a uniform random number?
- There are two main methods:
  1. Harvested natural randomness
  2. Pseudo-random number generation

Harvesting Natural Randomness

- The idea behind harvesting natural randomness is that there natural phenomenon that are sufficiently random, due to physical entropy, that we can treat them a random.

Harvesting Natural Randomness

- The idea behind harvesting natural randomness is that there natural phenomenon that are sufficiently random, due to physical entropy, that we can treat them a random.
- Examples of harvested sources of natural randomness include:

Harvesting Natural Randomness

- The idea behind harvesting natural randomness is that there natural phenomenon that are sufficiently random, due to physical entropy, that we can treat them a random.
- Examples of harvested sources of natural randomness include:
  - The amount of current flowing through an electrical circuit is actually a discrete number as it is caused by individual electrons moving through conductor. **Shot noise** is the random fluctuations in this discrete number.

Harvesting Natural Randomness

- The idea behind harvesting natural randomness is that there natural phenomenon that are sufficiently random, due to physical entropy, that we can treat them a random.
- Examples of harvested sources of natural randomness include:
    - The amount of current flowing through an electrical circuit is actually a discrete number as it is caused by individual electrons moving through conductor. **Shot noise** is the random fluctuations in this discrete number.
    - Source of nuclear radiation decay randomly and the amount of decay can be used as a source of randomness.

Harvesting Natural Randomness

- The idea behind harvesting natural randomness is that there natural phenomenon that are sufficiently random, due to physical entropy, that we can treat them a random.
- Examples of harvested sources of natural randomness include:
  - The amount of current flowing through an electrical circuit is actually a discrete number as it is caused by individual electrons moving through conductor. **Shot noise** is the random fluctuations in this discrete number.
  - Source of nuclear radiation decay randomly and the amount of decay can be used as a source of randomness.
  - When a beam of photons is directed at a semi-transparent mirror, the photons will randomly be reflected or transmitted.

Harvesting Natural Randomness

- The idea behind harvesting natural randomness is that there natural phenomenon that are sufficiently random, due to physical entropy, that we can treat them a random.
- Examples of harvested sources of natural randomness include:
  - The amount of current flowing through an electrical circuit is actually a discrete number as it is caused by individual electrons moving through conductor. **Shot noise** is the random fluctuations in this discrete number.
  - Source of nuclear radiation decay randomly and the amount of decay can be used as a source of randomness.
  - When a beam of photons is directed at a semi-transparent mirror, the photons will randomly be reflected or transmitted.
  - The temperature of a resistor in a circuit.

Harvesting Natural Randomness

- The idea behind harvesting natural randomness is that there natural phenomenon that are sufficiently random, due to physical entropy, that we can treat them a random.
- Examples of harvested sources of natural randomness include:
  - The amount of current flowing through an electrical circuit is actually a discrete number as it is caused by individual electrons moving through conductor. **Shot noise** is the random fluctuations in this discrete number.
  - Source of nuclear radiation decay randomly and the amount of decay can be used as a source of randomness.
  - When a beam of photons is directed at a semi-transparent mirror, the photons will randomly be reflected or transmitted.
  - The temperature of a resistor in a circuit.
  - Atmospheric noise detected by a radio receiver.

Harvesting Natural Randomness

- The idea behind harvesting natural randomness is that there natural phenomenon that are sufficiently random, due to physical entropy, that we can treat them a random.
- Examples of harvested sources of natural randomness include:
  - The amount of current flowing through an electrical circuit is actually a discrete number as it is caused by individual electrons moving through conductor. **Shot noise** is the random fluctuations in this discrete number.
  - Source of nuclear radiation decay randomly and the amount of decay can be used as a source of randomness.
  - When a beam of photons is directed at a semi-transparent mirror, the photons will randomly be reflected or transmitted.
  - The temperature of a resistor in a circuit.
  - Atmospheric noise detected by a radio receiver.
- The rate at which these can be harvested may be limited.

Pseudo-Random Numbers

- A **Pseudo-random number generator** returns a sequence of numbers that has the statistical properties of a uniform random sequence, but is actually deterministic given an initial value.

Pseudo-Random Numbers

- A **Pseudo-random number generator** returns a sequence of numbers that has the statistical properties of a uniform random sequence, but is actually deterministic given an initial value.
- This initial value is called a **seed**.

Pseudo-Random Numbers

- A **Pseudo-random number generator** returns a sequence of numbers that has the statistical properties of a uniform random sequence, but is actually deterministic given an initial value.
- This initial value is called a **seed**.
- Without knowledge of the seed, the sequence should look random.

Pseudo-Random Numbers

- A **Pseudo-random number generator** returns a sequence of numbers that has the statistical properties of a uniform random sequence, but is actually deterministic given an initial value.
- This initial value is called a **seed**.
- Without knowledge of the seed, the sequence should look random.
- Many such sequences exist.

Pseudo-Random Numbers

- A **Pseudo-random number generator** returns a sequence of numbers that has the statistical properties of a uniform random sequence, but is actually deterministic given an initial value.
- This initial value is called a **seed**.
- Without knowledge of the seed, the sequence should look random.
- Many such sequences exist.
- Given the seed, computing pseudo-random number is extremely fast.

Randomness in Numpy

`numpy.random` provides functions for generating pseudo-random samples from many common distributions.

Randomness in Numpy

`numpy.random` provides functions for generating pseudo-random samples from many common distributions.

```
>>> np.random.rand(5)
array([ 0.05321291,  0.7837193 ,  0.21213337,
    0.26499685,  0.4338131 ])
>>> np.random.randn(5)
array([-1.23805748, -0.18105506,  0.96736693,
    1.25355742, -0.08038946])
>>> np.random.randint(1,100,5)
array([28, 35, 32, 19, 94])
```

Sources of Randomness

Demo