



MITx CSE.0002x

Introduction to Computational Science and Engineering

[Help](#)

sandipan_dey ▾

- [Course](#)
- [Progress](#)
- [Dates](#)
- [Discussion](#)
- [MO Index](#)

[🏠](#) [Course](#) / [15 Fundamentals of Probability and Statisti...](#) / [15.2 Fundamentals of Probabil...](#)

15.2.2 Uniform Distribution

 [Bookmark this page](#)

MO2.3

MO2.13

In Chapter [14](#), we introduced probabilistic thinking using uniform distributions in which a parameter (i.e. a random variable) was assumed to be equally likely to occur between two limiting values $\mathbf{x_{min}}$ and $\mathbf{x_{max}}$. A sketch of the PDF for a uniform distribution is shown in Figure [15.2](#). For $\mathbf{x_{min}} < \mathbf{x} < \mathbf{x_{max}}$, the density has a constant value $\mathbf{f = h}$. The value of \mathbf{h} is not arbitrary, rather it is determined by Equation [15.4](#):

$$1 = \int_{-\infty}^{+\infty} f(\xi) \, d\xi = \int_{x_{\min}}^{x_{\max}} f(\xi) \, d\xi = h \int_{x_{\min}}^{x_{\max}} d\xi = h \times (x_{\max} - x_{\min}) \quad (15.6)$$

$$\Rightarrow h = 1/(x_{\max} - x_{\min}) \quad (15.7)$$

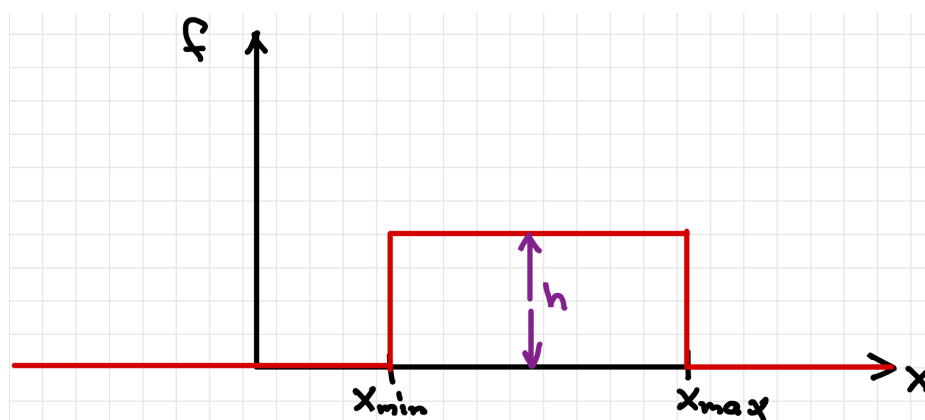


Figure 15.2: Uniform distribution. $h = 1 / (x_{\max} - x_{\min})$.

NumPy can be used to generate uniformly distributed random numbers by calling the `uniform` method in the Numpy default random number generator class:

```
import numpy as np

# xmin, xmax, and shape must be defined!

rng = np.random.default_rng()
```

[← Previous](#)

Next >



edX

- [About](#)
- [Affiliates](#)
- [edX for Business](#)
- [Open edX](#)
- [Careers](#)
- [News](#)

Legal

- [Terms of Service & Honor Code](#)
- [Privacy Policy](#)
- [Accessibility Policy](#)
- [Trademark Policy](#)
- [Sitemap](#)
- [Cookie Policy](#)
- [Your Privacy Choices](#)

Connect

- [Idea Hub](#)
- [Contact Us](#)
- [Help Center](#)
- [Security](#)
- [Media Kit](#)



© 2023 edX LLC. All rights reserved.
深圳市恒宇博科技有限公司 [粤ICP备17044299号-2](#)