Binomial Probability Distribution (discrete)

$$X \sim B(n, p)$$

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

$$\binom{a}{b} = \frac{a!}{b!(a-b)!}$$

$$E[X] = \mu = np$$

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

Notation in book and Python

Book	Python	
n	n	(total number of ”draws”)
p	p	(probability of success in each event)
x	k	(observed number of success’es, out of n possible)

Python Functions in `scipy.stats.binom`

- `rvs(n, p, size=...)`: Random variates
- `pmf(k, n, p)`: Probability mass function (book: $f(x)$)
- `cdf(k, n, p)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, n, p)`: Percent-point function (quantile)
- `mean(n, p)`: Mean
- `var(n, p)`: Variance
- `std(n, p)`: Standard deviation

Hypergeometric Distribution (discrete)

$$X \sim H(n, a, N)$$

$$f(x) = P(X = x) = \frac{\binom{a}{x}\binom{N-a}{n-x}}{\binom{N}{n}}$$

$$E[X] = \mu = n \cdot \frac{a}{N}$$

$$\text{Var}(X) = \sigma^2 = n \cdot \frac{a}{N} \cdot \frac{N-a}{N} \cdot \frac{N-n}{N-1}$$

Notation in Book and Python

Book	Python	
N	M	(total number of objects)
a	n	(total number of success objects)
n	N	(total number of ”draws”)
x	k	(observed number of success’es)

Python Functions in `scipy.stats.hypergeom`

- `rvs(M, n, N, size=...)`: Random variates
- `pmf(k, M, n, N)`: Probability mass function (book: $f(x)$)
- `cdf(k, M, n, N)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, M, n, N)`: Percent-point function (quantile)
- `mean(M, n, N)`: Mean
- `var(M, n, N)`: Variance
- `std(M, n, N)`: Standard deviation

Poisson Distribution (discrete)

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

$$f(x) = P(X = x) = \frac{\lambda^x}{x!} e^{-\lambda}$$

$$E[X] = \mu = \lambda$$

$$\text{Var}(X) = \sigma^2 = \lambda$$

Notation in Book and Python

Book	Python	
λ	`mu`	(average rate)
x	`k`	(observed number of events)

Python Functions in `scipy.stats.poisson`

- `rvs(mu, size=...)`: Random variates
- `pmf(k, mu)`: Probability mass function (book: $f(x)$)
- `cdf(k, mu)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, mu)`: Percent-point function (quantile)
- `mean(mu)`: Mean
- `var(mu)`: Variance
- `std(mu)`: Standard deviation

Normal Distribution (continuous)

$$X \sim \mathcal{N}(\mu, \sigma^2)$$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

$$E[X] = \mu$$

$$\mathrm{Var}(X) = \sigma^2$$

Notation in Book and Python

Book	Python	
μ	`loc`	(mean)
σ^2	`scale`	(std deviation)
x	x	(observed value)

Python Functions in `scipy.stats.norm`

- `rvs(loc, scale, size=...)`: Random variates
- `pdf(x, loc, scale)`: Probability density function (book: $f(x)$)
- `cdf(x, loc, scale)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, loc, scale)`: Percent-point function (quantile)
- `mean(loc, scale)`: Mean
- `var(loc, scale)`: Variance
- `std(loc, scale)`: Standard deviation

Uniform Distribution (continuous)

$$X \sim U(\alpha, \beta)$$

$$f(x) = \begin{cases} \frac{1}{\beta-\alpha} & \alpha \leq x \leq \beta \\ 0 & \text{otherwise} \end{cases}$$

$$E[X] = \mu = \frac{\alpha + \beta}{2}$$

$$\text{Var}(X) = \sigma^2 = \frac{(\beta - \alpha)^2}{12}$$

Notation in Book and Python

Book	Python	
α	`loc`	(lower bound)
β	`loc + scale`	(upper bound)
x	x	(oberserved value)

Python Functions in `scipy.stats.uniform`

- `rvs(loc, scale, size=...)`: Random variates
- `pdf(x, loc, scale)`: Probability density function (book: $f(x)$)
- `cdf(x, loc, scale)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, loc, scale)`: Percent-point function (quantile)
- `mean(loc, scale)`: Mean
- `var(loc, scale)`: Variance
- `std(loc, scale)`: Standard deviation

Exponential Distribution (continuous)

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

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$E[X] = \mu = \frac{1}{\lambda}$$

$$\text{Var}(X) = \sigma^2 = \frac{1}{\lambda^2}$$

Notation in Book and Python

Book	Python	
λ	`1/scale`	(rate)
	`loc`	(only use to shift x-axis)
$\mu = 1/\lambda$	`scale`	(average waiting time)
x	x	(observed value/waiting time between events)

Use: `loc` $= 0$ and `scale` $= 1/\lambda$ or `scale` $= \mu$

Python Functions in `scipy.stats.expon`

- `rvs(scale, size=...)`: Random variates
- `pdf(x, scale)`: Probability density function (book: $f(x)$)
- `cdf(x, scale)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, scale)`: Percent-point function (quantile)
- `mean(scale)`: Mean
- `var(scale)`: Variance
- `std(scale)`: Standard deviation

Lognormal Distribution (continuous)

$$X \sim \text{LN}(\alpha, \beta)$$

$$f(x) = \begin{cases} \frac{1}{x\sqrt{2\pi\beta^2}} \exp\left(-\frac{(\ln(x)-\alpha)^2}{2\beta^2}\right) & x > 0 \\ 0 & x \leq 0 \end{cases}$$

$$E[X] = e^{\alpha + \frac{\beta^2}{2}}$$

$$\text{Var}(X) = \left(e^{\beta^2} - 1\right) e^{2\alpha+\beta^2}$$

Notation in Book and Python

Book	Python	
α		(mean of $\ln(X)$)
β	`s`	(std of $\ln(X)$)
	`loc`	(only use to shift x-axis)
$\exp(\alpha)$	`scale`	
x	x	(observed value)

Use: `loc` $= 0$, `scale` $= \exp(\alpha)$ and `s` $= \beta$

Python Functions in `scipy.stats.lognorm`

- `rvs(s, loc, scale, size=...)`: Random variates (`s` corresponds to β)
- `pdf(x, s, loc, scale)`: Probability density function (book: $f(x)$)
- `cdf(x, s, loc, scale)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, s, loc, scale)`: Percent-point function (quantile)
- `mean(s, loc, scale)`: Mean
- `var(s, loc, scale)`: Variance
- `std(s, loc, scale)`: Standard deviation

t-Distribution (continuous)

$$X \sim t(\nu)$$

$$f(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Notation in Book and Python

Book	Python	
ν	`df`	(degrees of freedom)
x	x	(observed value)

Python Functions in `scipy.stats.t`

- `rvs(df, size=...)`: Random variates
- `pdf(x, df)`: Probability density function (book: $f(x)$)
- `cdf(x, df)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, df)`: Percent-point function (quantile)
- `mean(df)`: Mean
- `var(df)`: Variance
- `std(df)`: Standard deviation

F-Distribution (continuous)

$$X \sim F(\nu_1, \nu_2)$$

(see book def 2.95)

Notation in Book and Python

Book	Python	
ν_1	`dfn`	(numerator df)
ν_2	`dfd`	(denominator df)
x	x	(observed value)

Python Functions in `scipy.stats.f`

- `rvs(dfn, dfd, size=...)`: Random variates
- `pdf(x, dfn, dfd)`: Probability density function (book: $f(x)$)
- `cdf(x, dfn, dfd)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, dfn, dfd)`: Percent-point function (quantile)
- `mean(dfn, dfd)`: Mean
- `var(dfn, dfd)`: Variance
- `std(dfn, dfd)`: Standard deviation

Chi-Square Distribution (continuous)

$$X \sim \chi^2(\nu)$$

$$f(x) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} x^{\frac{\nu}{2}-1} e^{-x/2}, \quad x \geq 0$$

Notation in Book and Python

Book	Python	
ν	`df`	(degrees of freedom)
x	x	(observed value)

Python Functions in `scipy.stats.chi2`

- `rvs(df, size=...)`: Random variates
- `pdf(x, df)`: Probability density function (book: $f(x)$)
- `cdf(x, df)`: Cumulative distribution function (book: $F(x)$)
- `ppf(q, df)`: Percent-point function (quantile)
- `mean(df)`: Mean
- `var(df)`: Variance
- `std(df)`: Standard deviation