	R	Python	
Binomial	dbinom(x, size, prob)	from scipy.stats import binom binom.pmf(k, n, p)	probability mass function
	pbinom(q, size, prob)	binom.cdf(k, n, p)	cumulative distribution function
	qbinom(p, size, prob)	binom.ppf(q, n, p)	percent point function
	rbinom(n, size, prob)	import numpy as np **np**.random.binomial(n, p, size)	
Poisson	dpois(x, lambda)	from scipy.stats import poisson poisson.pmf(k, mu)	
	ppois(q, lambda)	poisson.cdf(k, mu)	
	qpois(p, lambda)	poisson.ppf(q, mu)	
	rpois(n, lambda)	**np**.random.poisson(lam, size)	
Geométrica	dgeom(x = # fracasos, prob)	from scipy.stats import geom geom.pmf(k = # intentos, p)	
	pgeom(q = # fracasos, prob)	geom.cdf(k = # intentos, p)	
	qgeom(p, prob)	geom.ppf(q, p)	
	rgeom(n, prob)	geom.rvs(p, size) **- 1**	
Binomial Negativa	dnbinom(x, size, prob)	from scipy.stats import nbinom nbinom.pmf(k, n, p)	
	pnbinom(q, size, prob)	nbinom.cdf(k, n, p)	
	qnbinom(p, size, prob)	nbinom.ppf(q, n, p)	
	rnbinom(n, size, prob)	nbinom.rvs(n, p, size)	
Exponencial	**d**exp(q, rate = lambda)	from scipy.stats import expon expon.p**df**(x , scale = 1/lambda)	densidad (no probabilidad) probability density function
	pexp(q, rate = lambda)	expon.cdf(x , scale = 1/lambda)	
	qexp(p, rate = lambda)	expon.ppf(q, scale = 1/lambda)	
	rexp(n , rate = lambda)	expon.rvs(size, scale=1/lambda)	
Normal	pnorm(q, mean, sd)	from scipy.stats import norm norm.cdf(x, loc = media, scale = desv)	
	qnorm(p, mean, sd)	norm.ppf(x, loc, scale)	
	rnorm(n, mean, sd)	np.random.normal(loc, scale, siz=10))	