

# Prob 140 Code Reference Sheet

## Distributions and their parameters

Discrete	Continuous	
bernoulli(p)	gamma(a, scale) [Where a=r, scale=1/lambda]	norm(loc, scale) [Where loc=mu, scale=sigma]
binom(n, p)		
geom(p) [On {k=0,1,...}]	chi2(df)	rayleigh(r)
hypergeom(M, n, N) [Where M=G]	expon(scale) [Where scale=1/lambda]	uniform(loc, scale) [Where loc=a, scale=b-a]
nbinom(n, p)		
poisson(mu)	beta(a, b) [Where a=r, b=s]	multivariate_norm al(mean, cov)

Example usage: stats.norm.pdf(x, mu, sigma)

## Useful scipy.stats functions

Name	Usage
pmf(k, *params)	Probability mass function (discrete)
pdf(x, *params)	Probability density function (continuous)
cdf(x, *params)	Cumulative density function
rvs(*params, size)	Samples from dist size times
ppf(q, *params)	Percentile point function (inverse of cdf)
mean(*params)	Mean of distribution
std(*params)	SD of distribution

## Assorted NumPy, SciPy, Matplotlib Functions

np.arange(start, stop, step), np.arange(start, stop), np.arange(stop)	Array of numbers starting at start (default 0), stopping at stop, with step size step (default 1).	plt.plot(x_values, y_values)	Plots a line graph connecting (x_1, y_1), ..., (x_n, y_n).
np.append(array, item)	Creates copy of array with item at end.	plt.scatter(x_values, y_values)	Plots a scatter plot for (x_1, y_1), ..., (x_n, y_n)
np.random.choice(array), np.random.choice(array, p)	Selects an item at random from array, weighted by probabilities array p.	scipy.special.comb(N, k)	N choose k
		scipy.special.factorial(n)	n!
		np.mean(array)	Mean of array.
		np.std(array)	SD of array.
		np.sort(array)	Returns sorted copy of array

## Datascience and prob140 functions

Note: tbl refers to a generic Table. dist refers to a 2-column table in which the probabilities sum to 1.

Table()	Creates an empty table.
tbl.with_column(n1, v1, n2, v2, ...)	Adds columns with the name n_i and values v_i.
tbl.scatter(x_column, y_column)	Draws a scatter plot consisting of one point for each row of the table.
tbl.hist(column, bins)	Generates a histogram of the numerical values in a column.
tbl.apply(function, column)	Returns an array where a function is applied to each item in a column.
dist.values(values)	Adds a column with values of probability distribution.
dist.probability(probabilities)	Adds a column of probabilities corresponding to existing values.
dist.probability_function(f)	Adds a column of probabilities by applying function to existing values.
dist.ev()	Finds the expected value of the distribution.
dist.sd()	Finds the sd of the distribution.
dist.sample_from_dist(n=1)	Samples n values independently from the distribution.
emp_dist(array)	Returns the empirical distribution of the array.

## Prob140 Plotting functions

Plot(dist)	Plots the histogram for a discrete distribution.
Plots(label1, dist1, label2, dist2, ...)	Plots multiple discrete probability distributions.
Plot_3d(x_limits, y_limits, f)	Plots the 3D joint density defined by f(x, y).
Plot_bivariate_normal(mu, cov)	Plots the joint density of a bivariate normal distribution with mean vector mu and covariance matrix cov.
Scatter_multivariate_normal(mu, cov, n)	Plots a point cloud of n points drawn from a trivariate normal distribution with mean vector mu and covariance matrix cov.
multivariate_normal_regression(mu, cov, n)	Plots a point cloud for random vector [Y, X1, X2]^T drawn from a trivariate normal distribution with mean vector mu and covariance matrix cov. Also draws the true regression plane.





