powerlaw Documentation

Release 0.1

Shagun Sodhani

Sep 23, 2017

Contents

Contents:

CHAPTER 1

powerlaw

powerlaw package

Submodules

powerlaw.distribution module

powerlaw.distribution.**exponential_series**(*Lambda=1.0*, *n=1*, *xmin=1.0*, *discrete=False*)

Generator to generate a stream of numbers taken from exponential distribution.

Parameters

Lambda [Float/Integer, Lambda constant for the distribution.] Default value is 1.0

n [Integer, number of elements to be generated.] Default value is 1 If n < 0, then unbounded number of elements are generated.

xmin [Float/Integer, xmin for the distribution.] Default value is 1.0

discrete [Boolean, Whether the distribution is to be discrete or not (continous).] Default value is False

powerlaw.distribution.**frequency_distribution**(*series*, *pdf=True*, *ccdf=True*)

Generator to generates pdf(probability distribution function) or cdf(cummulative distribution function) or ccdf(complementary cummulative distribution function) of any series.

Parameters

series : list of values.

pdf [Boolean. If True, return pdf else cdf] Default value is True

ccdf [Boolean. This is considered only if pdf is set to False. If ccdf = True, return ccdf else cdf] Default value is True

Returns

(key, value) pairs are returned where key is one of the entries from the input series and value is the corresponding pdf for the key. The pairs are sorted by key.

powerlaw.distribution.**plot_pdf_series**(*series*)
Plots pdf(probability distribution function) for any series.

Parameters series : list of values.

Returns Log log plot of the values.

powerlaw.distribution.**powerlaw_series**(*Alpha=2.0*, *n=1*, *xmin=1.0*, *discrete=False*)
Generator to generate a stream of numbers taken from powerlaw distribution.

Parameters

Alpha [Float/Integer, Alpha constant for the distribution.] Default value is 2.0 Alpha value should be greater than 1.0

n [Integer, number of elements to be generated.] Default value is 1 If n < 0, then unbounded number of elements are generated.

xmin [Float/Integer, xmin for the distribution.] Default value is 1.0

discrete [Boolean, Whether the distribution is to be discrete or not (continous).] Default value is False

powerlaw.distribution.**random**() → x in the interval [0, 1).

powerlaw.distribution.**random_series**(*n=1*)
Generator to generate a stream of random numbers.

Parameters

n [Integer, number of elements to be generated.] Default value is 1 If n < 0, then unbounded number of elements are generated.

powerlaw.distribution.**stretched_exponential_series**(*Lambda=1.0*, *Beta=1.0*, *n=1*, *xmin=1.0*, *discrete=False*)
Generator to generate a stream of numbers taken from stretched exponential distribution.

Parameters

Lambda [Float/Integer, Lambda constant for the distribution.] Default value is 1.0

Beta [Float/Integer, Beta constant for the distribution.] Default value is 1.0

n [Integer, number of elements to be generated.] Default value is 1 If n < 0, then unbounded number of elements are generated.

xmin [Float/Integer, xmin for the distribution.] Default value is 1.0

discrete [Boolean, Whether the distribution is to be discrete or not (continous).] Default value is False

powerlaw.regression module

powerlaw.regression.**estimate_parameters**(*series*, *min_size_series=50*, *discrete=False*)
Apply Clauset et al.'s method to find the best fit value of xmin and Alpha.

Parameters

series : series of data to be fit.

min_size_series : Minimum possible size of the distribution to which power-law fit will be attempted. Fitting power-law to a very small series would give biased results where power-law may appear to be

a good fit even when data is not drawn from power-law distribution. The default value is taken to be 50 as suggested in the paper.

discrete : Boolean, whether to treat series as discrete or continous. Default value is False

Returns

Tuple of (Estimated xmin, Estimated Alpha value, minimum KS statistics score).

powerlaw.regression.**estimate_scaling_parameter**(*series*, *xmin=1*, *discrete=False*)

Perform Method of Maximum Liklihood (MLE) to find the best fit value of Alpha.

Parameters

series : series of data to be fit. xmin : Float/Integer, xmin for the distribution - assumed to be known before-hand. Default value is 1.0 discrete : Boolean, whether to treat series as discrete or continous. Default value is False.

Returns

Estimated Alpha value.

powerlaw.regression.**generate_dataset**(*series*, *xmin*, *alpha*, *epsilon=0.01*)

Generator to generate datasets for goodness_of_fit test.

Parameters

series : series of data on which the power-law model was fitted.

xmin : xmin for the fitted power-law model.

alpha : alpha for the fitted power-law model.

epsilon : desired accuracy in p-value. Default is set to 0.01

Returns

A generator to generate list of numbers (datasets).

powerlaw.regression.**goodness_of_fit**(*series*, *xmin*, *alpha*, *ks_statistics*, *epsilon=0.01*, *min_size_series=50*)

Function to calculate the p-value as a measure of goodness_of_fit for the fitted model.

Parameters

series : series of data on which the power-law model was fitted.

xmin : xmin for the fitted power-law model.

alpha : alpha for the fitted power-law model.

ks_statistics : KS statistics for the fitted power-law model.

epsilon : desired accuracy in p-value. Default is set to 0.01.

min_size_series : Minimum possible size of the distribution to which power-law fit will be attempted. This value is used when fitting power-law to the generated datasets. The default value is taken to be 50. For further details, see *estimate_parameters()*.

Returns

p-value for the fitted model.

powerlaw.regression.**least_square_regression**(*x*, *y*, *xlabel='x'*, *ylabel='y'*, *prefix=''*, *suffix=''*)

Perform least square regression to find the best fit line and returns the slope of the line.

Parameters x : List of values along x axis. y : List of values along y axis.

powerlaw.utils module

powerlaw.utils.**unique**(*series*)

Generator to generate all unique values from the input series, in the order they appear for the first time.

Parameters

series : Input series.

Module contents

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

Index