benford$_{py}$

Release 0.3.3

Jun 29, 2021

Contents:

CHAPTER 1

benford

1.1 benford package

1.1.1 benford.benford module

class benford.benford.**Base**(*data*, *decimals*, *sign='all'*, *sec_order=False*)

Bases: pandas.core.frame.DataFrame

Internalizes and prepares the data for Analysis.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.'

Raises TypeError – if not receiving *int* or *float* as input.

class benford.benford.**Test**(*base*, *digs*, *confidence*, *limit_N=None*, *sec_order=False*)

Bases: pandas.core.frame.DataFrame

Transforms the original number sequence into a DataFrame reduced by the ocurrences of the chosen digits, creating other computed columns

Parameters

- **base** – The Base object with the data prepared for Analysis
- **digs** – Tells which test to perform: 1: first digit; 2: first two digits; 3: furst three digits; 22: second digit; -2: last two digits.

- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show.
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.

N
Number of records in the sample to consider in computations

ddf
Degrees of Freedom to look up for the critical chi-square value

chi_square
Chi-square statistic for the given test

KS
Kolmogorov-Smirnov statistic for the given test

MAD
Mean Absolute Deviation for the given test

confidence
Confidence level to consider when setting some critical values

digs
numerical representation of the test at hand. 1: F1D; 2: F2D; 3: F3D; 22: SD; -2: L2D.

Type int

sec_order
True if the test is a Second Order one

Type bool

update_confidence(*new_conf*, *check=True*)
Sets a new confidence level for the Benford object, so as to be used to produce critical values for the tests

Parameters

- **new_conf** – new confidence level to draw lower and upper limits when plotting and to limit the top deviations to show, as well as to calculate critical values for the tests' statistics.
- **check** – checks the value provided for the confidence. Defaults to True

critical_values
a dictionary with the critical values for the test at hand, according to the current confidence level.

Type dict

show_plot(*save_plot=None*, *save_plot_kwargs=None*)
Draws the test plot.

Parameters

- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when save_plot is a string with the figure file path/name.

report (*high_Z='pos'*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*)
Handles the report especific to the test, considering its statistics and according to the current confidence level.

Parameters

- **high_Z** (*int*) – chooses which Z scores to be used when displaying results, according to the confidence level chosen. Defaluts to 'pos', which will highlight only values higher than the expexted frequencies; 'all' will highlight both extremes (positive and negative); and an integer, which will use the first n entries, positive and negative, regardless of whether Z is higher than the critical value or not.
- **show_plot** – calls the show_plot method, to draw the test plot
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

class benford.benford.**Summ**(*base*, *test*)
Bases: pandas.core.frame.DataFrame

Gets the base object and outputs a Summation test object

Parameters

- **base** – The Base object with the data prepared for Analysis
- **test** – The test for which to compute the summation

MAD = None
Mean Absolute Deviation for the test

confidence = None
Confidence level to consider when setting some critical values

show_plot (*save_plot=None*, *save_plot_kwargs=None*)
Draws the Summation test plot

Parameters

- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when save_plot is a string with the figure file path/name.

report (*high_diff=None*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*)
Gives the report on the Summation test.

Parameters

- **high_diff** – Number of records to show after ordering by the absolute differences between the found and the expected proportions
- **show_plot** – calls the show_plot method, to draw the Summation test plot

- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

class benford.benford.**Mantissas**(*data*, *confidence=95*, *limit_N=None*)

Bases: object

Computes and holds the mantissas of the logarithms of the records

Parameters

- **data** – sequence to compute mantissas from. numpy 1D array, pandas Series of pandas DataFrame column.
- **confidence** – confidence level for computing the critical values to compare with some statistics

data = None

pandas DataFrame with the mantissas

Type (DataFrame)

stats

update_confidence(*new_conf*, *check=True*)

Sets a new confidence level for the Benford object, so as to be used to produce critical values for the tests

Parameters

- **new_conf** – new confidence level to draw lower and upper limits when plotting and to limit the top deviations to show, as well as to calculate critical values for the tests' statistics.
- **check** – checks the value provided for the confidence. Defaults to True

report(*show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*)

Displays the Mantissas test stats.

Parameters

- **show_plot** – shows the Ordered Mantissas plot and the Arc Test plot. Defaults to True.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

show_plot(*figsize=(12, 6)*, *save_plot=None*, *save_plot_kwargs=None*)

Plots the ordered mantissas and a line with the expected inclination.

Parameters

- **figsize** (*tuple*) – figure size dimensions
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.

- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when save_plot is a string with the figure file path/name.

arc_test(*grid=True*, *figsize=12*, *save_plot=None*, *save_plot_kwargs=None*)

Adds two columns to Mantissas's DataFrame equal to their "X" and "Y" coordinates, plots its to a scatter plot and calculates the gravity center of the circle.

Parameters

- **grid** – show grid of the plot. Defaluts to True.
- **figsize** (*int*) – size of the figure to be displayed. Since it is a square, there is no need to provide a tuple, like is usually the case with matplotlib.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

class benford.benford.**Benford**(*data*, *decimals=2*, *sign='all'*, *confidence=95*, *mantissas=True*, *sec_order=False*, *summation=False*, *limit_N=None*, *verbose=True*)

Bases: object

Initializes a Benford Analysis object and computes the proportions for the digits. The tets dataFrames are atributes, i.e., obj.F1D is the First Digit DataFrame, the obj.F2D,the First Two Digits one, and so one, F3D for First Three Digits, SD for Second Digit and L2D for Last Two Digits.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a tuple with a pandas DataFrame and the name (str) of the chosen column. Values must be integers or floats.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.
- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show, as well as to calculate critical values for the tests' statistics. Defaults to 95.
- **mantissas** (*bool*) – opts for also running the mantissas Test. Defaulst to True
- **sec_order** – runs the Second Order tests, which are the Benford's tests performed on the differences between the ordered sample (a value minus the one before it, and so on). If the original series is Benford- compliant, this new sequence should aldo follow Beford. The Second Order can also be called separately, through the method sec_order().
- **summation** – creates the Summation DataFrames for the First, First Two, and First Three Digits. The summation tests can also be called separately, through the method summation().
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.

- **verbose** – gives some information about the data and the registries used and discarded for each test.

data
the raw data provided for the analysis

chosen
the column of the DataFrame to be analysed or the data itself

sign
which number sign(s) to include in the analysis

Type str

confidence
current confidence level

limit_N
sample size to use in computations

Type int

verbose
verbose or not

Type bool

base
the Base, pre-processed object

tests
keeps track of the tests the instance has

Type `list` of `str`

update_confidence(*new_conf*, *tests=None*)
Sets (a) new confidence level(s) for the Benford object, so as to be used to produce critical values for the tests.

Parameters

- **new_conf** – new confidence level to draw lower and upper limits when plotting and to limit the top deviations to show, as well as to calculate critical values for the tests' statistics.
- **tests** (`list` of `str`) – list of tests names (strings) to have their confidence updated. If only one, provide a one-element list, like ['F1D']. Defauts to None, in which case it will use the instance .test list attribute.

Raises `ValueError` – if the test argument is not a *list* or *None*.

all_confidences
a dictionary with a confidence level for each computed tests, when applicable.

Type dict

mantissas()
Adds a Mantissas object to the tests, with all its statistics and plotting capabilities.

sec_order()
Runs the Second Order tests, which are the Benford's tests performed on the differences between the ordered sample (a value minus the one before it, and so on). If the original series is Benford- compliant, this new sequence should aldo follow Beford. The Second Order can also be called separately, through the method sec_order().

summation()
Creates Summation test DataFrames from Base object

class benford.benford.**Source**(*data*, *decimals=2*, *sign='all'*, *sec_order=False*, *verbose=True*, *inform=None*)
Bases: pandas.core.frame.DataFrame

Prepares the data for Analysis. pandas DataFrame subclass.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.
- **sec_order** – choice for the Second Order Test, which cumputes the differences between the ordered entries before running the Tests.
- **verbose** (*bool*) – tells the number of registries that are being subjected to the analysis; defaults to True.

Raises

- ValueError – if the *sign* arg is not in ['all', 'pos', 'neg']
- TypeError – if not receiving *int* or *float* as input.

verbose = None
verbose or not

Type (bool)

mantissas(*report=True*, *show_plot=True*, *figsize=(15, 8)*, *save_plot=None*, *save_plot_kwargs=None*)
Calculates the mantissas, their mean and variance, and compares them with the mean and variance of a Benford's sequence.

Parameters

- **report** – prints the mamtissas mean, variance, skewness and kurtosis for the sequence studied, along with reference values.
- **show_plot** – plots the ordered mantissas and a line with the expected inclination. Defaults to True.
- **figsize** – tuple that sets the figure dimensions.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

first_digits(*digs*, *confidence=None*, *high_Z='pos'*, *limit_N=None*, *MAD=False*, *MSE=False*, *chi_square=False*, *KS=False*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *simple=False*, *bhat_coeff=False*, *bhat_dist=False*, *kl_diverg=False*, *ret_df=False*)

Performs the Benford First Digits test with the series of numbers provided, and populates the mapping dict for future selection of the original series.

Parameters

- **digs** (*int*) – number of first digits to consider. Must be 1 (first digit), 2 (first two digits) or 3 (first three digits).
- **verbose** (*bool*) – tells the number of registries that are being subjected to the analysis; defaults to True
- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show, as well as to calculate critical values for the tests' statistics. Defaults to None.
- **high_Z** (*int*) – chooses which Z scores to be used when displaying results, according to the confidence level chosen. Defaluts to 'pos', which will highlight only values higher than the expexted frequencies; 'all' will highlight both extremes (positive and negative); and an integer, which will use the first n entries, positive and negative, regardless of whether Z is higher than the confidence or not.
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.
- **MAD** (*bool*) – calculates the Mean Absolute Difference between the found and the expected distributions; defaults to False.
- **MSE** (*bool*) – calculates the Mean Square Error of the sample; defaults to False.
- **bhat_coeff** (*bool*) – computes the Bhattacharyya Coefficient between the found and the expected (Benford) digits distribution; defaults to Fasle
- **bhat_dist** (*bool*) – calculates the Bhattacharyya Distance between the found and the expected (Benford) digits distribution; defaults to Fasle
- **kl_diverg** (*bool*) – calculates the Kulback-Laibler Divergence between the found and the expected (Benford) digits distribution; defaults to False
- **show_plot** (*bool*) – draws the test plot. Defaults to True.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.
- **ret_df** – returns the test DataFrame. Defaults to False. True if run by the test function.

Returns

DataFrame with the Expected and Found proportions, and the Z scores of the differences

second_digit(*confidence=None*, *high_Z='pos'*, *limit_N=None*, *MAD=False*, *MSE=False*, *chi_square=False*, *KS=False*, *bhat_coeff=False*, *bhat_dist=False*, *kl_diverg=False*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *simple=False*, *ret_df=False*)

Performs the Benford Second Digit test with the series of numbers provided.

Parameters

- **verbose** (*bool*) – tells the number of registries that are being subjected to the analysis; defaults to True
- **MAD** (*bool*) – calculates the Mean Absolute Difference between the found and the expected distributions; defaults to False.
- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show, as well as to calculate critical values for the tests' statistics. Defaults to None.
- **high_Z** (*int*) – chooses which Z scores to be used when displaying results, according to the confidence level chosen. Defaluts to 'pos', which will highlight only values higher than the expexted frequencies; 'all' will highlight both extremes (positive and negative); and an integer, which will use the first n entries, positive and negative, regardless of whether Z is higher than the confidence or not.
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.
- **MSE** (*bool*) – calculates the Mean Square Error of the sample; defaults to False.
- **bhat_coeff** (*bool*) – computes the Bhattacharyya Coefficient between the found and the expected (Benford) digits distribution; defaults to Fasle
- **bhat_dist** (*bool*) – calculates the Bhattacharyya Distance between the found and the expected (Benford) digits distribution; defaults to Fasle
- **kl_diverg** (*bool*) – calculates the Kulback-Laibler Divergence between the found and the expected (Benford) digits distribution; defaults to False
- **show_plot** (*bool*) – draws the test plot.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.
- **ret_df** – returns the test DataFrame. Defaults to False. True if run by the test function.

Returns

DataFrame with the Expected and Found proportions, and the Z scores of the differences

last_two_digits(*confidence=None*, *high_Z='pos'*, *limit_N=None*, *MAD=False*, *MSE=False*, *chi_square=False*, *KS=False*, *bhat_coeff=False*, *bhat_dist=False*, *kl_diverg=False*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *simple=False*, *ret_df=False*)

Performs the Benford Last Two Digits test with the series of numbers provided.

Parameters

- **verbose** (*bool*) – tells the number of registries that are being subjected to the analysis; defaults to True
- **MAD** (*bool*) – calculates the Mean Absolute Difference between the found and the expected distributions; defaults to False.
- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show, as well as to calculate critical values for the tests' statistics. Defaults to None.

- **high_Z** (*int*) – chooses which Z scores to be used when displaying results, according to the confidence level chosen. Defaluts to 'pos', which will highlight only values higher than the expexted frequencies; 'all' will highlight both extremes (positive and negative); and an integer, which will use the first n entries, positive and negative, regardless of whether Z is higher than the confidence or not.
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.
- **MSE** (*bool*) – calculates the Mean Square Error of the sample; defaults to False.
- **bhat_coeff** (*bool*) – computes the Bhattacharyya Coefficient between the found and the expected (Benford) digits distribution; defaults to Fasle
- **bhat_dist** (*bool*) – calculates the Bhattacharyya Distance between the found and the expected (Benford) digits distribution; defaults to Fasle
- **kl_diverg** (*bool*) – calculates the Kulback-Laibler Divergence between the found and the expected (Benford) digits distribution; defaults to False
- **show_plot** (*bool*) – draws the test plot.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns

DataFrame with the Expected and Found proportions, and the Z scores of the differences

summation(*digs=2*, *top=20*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *ret_df=False*)

Performs the Summation test. In a Benford series, the sums of the entries begining with the same digits tends to be the same.

Parameters

- **digs** – tells the first digits to use. 1- first; 2- first two; 3- first three. Defaults to 2.
- **top** – choses how many top values to show. Defaults to 20.
- **show_plot** – plots the results. Defaults to True.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns

DataFrame with the Expected and Found proportions, and their absolute differences

duplicates(*top_Rep=20*, *inform=None*)

Performs a duplicates test and maps the duplicates count in descending order.

Parameters

- **verbose** (*bool*) – tells how many duplicated entries were found and prints the top numbers according to the top_Rep argument. Defaluts to True.
- **top_Rep** – int or None. Chooses how many duplicated entries will be shown withe the top repititions. Defaluts to 20. If None, returns al the ordered repetitions.

Returns

DataFrame with the duplicated records and their occurrence counts, in descending order (if verbose is False; if True, prints to terminal).

Raises `ValueError` – if the *top_Rep* arg is not int or None.

class benford.benford.**Roll_mad**(*data*, *test*, *window*, *decimals=2*, *sign='all'*)

Bases: `object`

Applies the MAD to sequential subsets of the Series, returning another Series.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **test** – tells which test to use. 1: Fisrt Digits; 2: First Two Digits; 3: First Three Digits; 22: Second Digit; and -2: Last Two Digits.
- **window** – size of the subset to be used.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.

test = None

the test (F1D, SD, F2D...) used for the MAD calculation and critical values

show_plot(*figsize=(15*, *8)*, *save_plot=None*, *save_plot_kwargs=None*)

Shows the rolling MAD plot

Parameters

- **figsize** – the figure dimensions.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when save_plot is a string with the figure file path/name.

class benford.benford.**Roll_mse**(*data*, *test*, *window*, *decimals=2*, *sign='all'*)

Bases: `object`

Applies the MSE to sequential subsets of the Series, returning another Series.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **test** – tells which test to use. 1: Fisrt Digits; 2: First Two Digits; 3: First Three Digits; 22: Second Digit; and -2: Last Two Digits.

- **window** – size of the subset to be used. decimals: number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. 'pos': only the positive entries; 'neg': only negative entries; 'all': all entries but zeros. Defaults to 'all'.

show_plot (*figsize=(15, 8)*, *save_plot=None*, *save_plot_kwargs=None*)
Shows the rolling MSE plot

Parameters

- **figsize** – the figure dimensions.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when save_plot is a string with the figure file path/name.

benford.benford.**first_digits**(*data*, *digs*, *decimals=2*, *sign='all'*, *verbose=True*, *confidence=None*, *high_Z='pos'*, *limit_N=None*, *MAD=False*, *MSE=False*, *chi_square=False*, *KS=False*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *inform=None*)
Performs the Benford First Digits test on the series of numbers provided.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. 'pos': only the positive entries; 'neg': only negative entries; 'all': all entries but zeros. Defaults to 'all'.
- **digs** (*int*) – number of first digits to consider. Must be 1 (first digit), 2 (first two digits) or 3 (first three digits).
- **verbose** (*bool*) – tells the number of registries that are being subjected to the analysis and returns tha analysis DataFrame sorted by the highest Z score down. Defaults to True.
- **MAD** (*bool*) – calculates the Mean Absolute Difference between the found and the expected distributions; defaults to False.
- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show. Defaults to None.
- **high_Z** (*int*) – chooses which Z scores to be used when displaying results, according to the confidence level chosen. Defaluts to 'pos', which will highlight only values higher than the expexted frequencies; 'all' will highlight both extremes (positive and negative); and an integer, which will use the first n entries, positive and negative, regardless of whether Z is higher than the confidence or not.
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.
- **MSE** (*bool*) – calculates the Mean Square Error of the sample; defaults to False.

- **chi_square** – calculates the chi_square statistic of the sample and compares it with a critical value, according to the confidence level chosen and the series's degrees of freedom. Defaults to False. Requires confidence != None.
- **KS** – calculates the Kolmogorov-Smirnov test, comparing the cumulative distribution of the sample with the Benford's, according to the confidence level chosen. Defaults to False. Requires confidence != None.
- **show_plot** (*bool*) – draws the test plot.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns

DataFrame with the Expected and Found proportions, and the Z scores of the differences if the confidence is not None.

benford.benford.**second_digit**(*data*, *decimals=2*, *sign='all'*, *verbose=True*, *confidence=None*, *high_Z='pos'*, *limit_N=None*, *MAD=False*, *MSE=False*, *chi_square=False*, *KS=False*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *inform=None*)

Performs the Benford Second Digits test on the series of numbers provided.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. 'pos': only the positive entries; 'neg': only negative entries; 'all': all entries but zeros. Defaults to 'all'.
- **verbose** (*bool*) – tells the number of registries that are being subjected to the analysis and returns tha analysis DataFrame sorted by the highest Z score down. Defaults to True.
- **MAD** (*bool*) – calculates the Mean Absolute Difference between the found and the expected distributions; defaults to False.
- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show. Defaults to None.
- **high_Z** (*int*) – chooses which Z scores to be used when displaying results, according to the confidence level chosen. Defaluts to 'pos', which will highlight only values higher than the expexted frequencies; 'all' will highlight both extremes (positive and negative); and an integer, which will use the first n entries, positive and negative, regardless of whether Z is higher than the confidence or not.
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.
- **MSE** (*bool*) – calculates the Mean Square Error of the sample; defaults to False.

- **chi_square** – calculates the chi_square statistic of the sample and compares it with a critical value, according to the confidence level chosen and the series's degrees of freedom. Defaults to False. Requires confidence != None.
- **KS** – calculates the Kolmogorov-Smirnov test, comparing the cumulative distribution of the sample with the Benford's, according to the confidence level chosen. Defaults to False. Requires confidence != None.
- **show_plot** (*bool*) – draws the test plot.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns

DataFrame with the Expected and Found proportions, and the Z scores of the differences if the confidence is not None.

benford.benford.**last_two_digits**(*data*, *decimals=2*, *sign='all'*, *verbose=True*, *confidence=None*, *high_Z='pos'*, *limit_N=None*, *MAD=False*, *MSE=False*, *chi_square=False*, *KS=False*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *inform=None*)

Performs the Last Two Digits test on the series of numbers provided.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column,with values being integers or floats.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. 'pos': only the positive entries; 'neg': only negative entries; 'all': all entries but zeros. Defaults to 'all'.
- **verbose** (*bool*) – tells the number of registries that are being subjected to the analysis and returns tha analysis DataFrame sorted by the highest Z score down. Defaults to True.
- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show. Defaults to None.
- **high_Z** (*int*) – chooses which Z scores to be used when displaying results, according to the confidence level chosen. Defaluts to 'pos', which will highlight only values higher than the expexted frequencies; 'all' will highlight both extremes (positive and negative); and an integer, which will use the first n entries, positive and negative, regardless of whether Z is higher than the confidence or not.
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.
- **MAD** (*bool*) – calculates the Mean Absolute Difference between the found and the expected distributions; defaults to False.
- **MSE** (*bool*) – calculates the Mean Square Error of the sample; defaults to False.

- **chi_square** – calculates the chi_square statistic of the sample and compares it with a critical value, according to the confidence level chosen and the series's degrees of freedom. Defaults to False. Requires confidence != None.
- **KS** – calculates the Kolmogorov-Smirnov test, comparing the cumulative distribution of the sample with the Benford's, according to the confidence level chosen. Defaults to False. Requires confidence != None.
- **show_plot** (*bool*) – draws the test plot.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns

DataFrame with the Expected and Found proportions, and the Z scores of the differences if the confidence is not None.

benford.benford.**mantissas**(*data*, *report=True*, *show_plot=True*, *arc_test=True*, *save_plot=None*, *save_plot_kwargs=None*, *inform=None*)

Extraxts the mantissas of the records logarithms

Parameters

- **data** – sequence to compute mantissas from, numpy 1D array, pandas Series of pandas DataFrame column.
- **report** – prints the mamtissas mean, variance, skewness and kurtosis for the sequence studied, along with reference values.
- **show_plot** – plots the ordered mantissas and a line with the expected inclination. Defaults to True.
- **arc_test** – draws the Arc Test plot. Defaluts to True.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns Series with the data mantissas.

benford.benford.**summation**(*data*, *digs=2*, *decimals=2*, *sign='all'*, *top=20*, *verbose=True*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *inform=None*)

Performs the Summation test. In a Benford series, the sums of the entries begining with the same digits tends to be the same. Works only with the First Digits (1, 2 or 3) test.

Parameters

- **digs** – tells the first digits to use: 1- first; 2- first two; 3- first three. Defaults to 2.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.

- **top** – choses how many top values to show. Defaults to 20.
- **show_plot** – plots the results. Defaults to True.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns

DataFrame with the Summation test, whether sorted in descending order (if verbose == True) or not.

benford.benford.**mad**(*data*, *test*, *decimals=2*, *sign='all'*, *verbose=False*)

Calculates the Mean Absolute Deviation of the Series

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **test** – informs which base test to use for the mad.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.

Returns the Mean Absolute Deviation of the Series

Return type float

benford.benford.**mse**(*data*, *test*, *decimals=2*, *sign='all'*, *verbose=False*)

Calculates the Mean Squared Error of the Series

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **test** – informs which base test to use for the mad.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.

Returns the Mean Squared Error of the Series

Return type float

benford.benford.**bhattacharyya_distance**(*data*, *test*, *decimals*, *sign='all'*, *verbose=False*)

Computes the Bhattacharyya Distance between the Found and the Expected (Benford) digits distributions, according toe the test chosen (First, Second, First Two...)

Parameters

- **data** (*ndarray, Series*) – sequence to be evaluated, with values being integers or floats.
- **test** (*int, str*) – informs which base test to be used.
- **decimals** (*int*) – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** (*str, optional*) – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to "all".

Returns the Bhattacharyya Distance between the distributions

Return type float

benford.benford.**kullback_leibler_divergence**(*data*, *test*, *decimals*, *sign='all'*, *verbose=False*)

Computes the Kulback-Leibler Divergence between the Found and the Expected (Benford) digits distributions, according toe the test chosen (First, Second, First Two...).

Parameters

- **data** (*ndarray, Series*) – sequence to be evaluated, with values being integers or floats.
- **test** (*int, str*) – informs which base test to be used.
- **decimals** (*int*) – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** (*str, optional*) – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to "all".

Returns the Kulback-Leibler Divergence between the distributions

Return type float

benford.benford.**mad_summ**(*data*, *test*, *decimals=2*, *sign='all'*, *verbose=False*)

Calculate the Mean Absolute Deviation of the Summation Test

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **test** – informs which base test to use for the summation mad.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.

Returns the Mean Absolute Deviation of the Summation Test

Return type float

benford.benford.**rolling_mad**(*data*, *test*, *window*, *decimals=2*, *sign='all'*, *show_plot=False*, *save_plot=None*, *save_plot_kwargs=None*)

Applies the MAD to sequential subsets of the records.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **test** – tells which test to use. 1: Fisrt Digits; 2: First Two Digits; 3: First Three Digits; 22: Second Digit; and -2: Last Two Digits.
- **window** – size of the subset to be used.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.
- **show_plot** (*bool*) – draws the test plot.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns Series with sequentially computed MADs.

benford.benford.**rolling_mse**(*data*, *test*, *window*, *decimals=2*, *sign='all'*, *show_plot=False*, *save_plot=None*, *save_plot_kwargs=None*)

Applies the MSE to sequential subsets of the records.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **test** – tells which test to use. 1: Fisrt Digits; 2: First Two Digits; 3: First Three Digits; 22: Second Digit; and -2: Last Two Digits.
- **window** – size of the subset to be used.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.
- **show_plot** (*bool*) – draws the test plot.
- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns Series with sequentially computed MSEs.

benford.benford.**duplicates**(*data*, *top_Rep=20*, *verbose=True*, *inform=None*)

Performs a duplicates test and maps the duplicates count in descending order.

Parameters

- **data** – sequence to take the duplicates from. pandas Series or numpy Ndarray.
- **verbose** (*bool*) – tells how many duplicated entries were found and prints the top numbers according to the top_Rep argument. Defaluts to True.
- **top_Rep** – chooses how many duplicated entries will be shown withe the top repititions. int or None. Defaluts to 20. If None, returns al the ordered repetitions.

Returns DataFrame with the duplicated records and their respective counts

Raises `ValueError` – if the *top_Rep* arg is not int or None.

benford.benford.**second_order**(*data*, *test*, *decimals=2*, *sign='all'*, *verbose=True*, *MAD=False*, *confidence=None*, *high_Z='pos'*, *limit_N=None*, *MSE=False*, *show_plot=True*, *save_plot=None*, *save_plot_kwargs=None*, *inform=None*)

Performs the chosen test after subtracting the ordered sequence by itself. Hence Second Order.

Parameters

- **data** – sequence of numbers to be evaluated. Must be a numpy 1D array, a pandas Series or a pandas DataFrame column, with values being integers or floats.
- **test** – the test to be performed - 1 or 'F1D': First Digit; 2 or 'F2D': First Two Digits; 3 or 'F3D': First three Digits; 22 or 'SD': Second Digits; -2 or 'L2D': Last Two Digits.
- **decimals** – number of decimal places to consider. Defaluts to 2. If integers, set to 0. If set to -infer-, it will remove the zeros and consider up to the fifth decimal place to the right, but will loose performance.
- **sign** – tells which portion of the data to consider. pos: only the positive entries; neg: only negative entries; all: all entries but zeros. Defaults to all.
- **verbose** (*bool*) – tells the number of registries that are being subjected to the analysis and returns tha analysis DataFrame sorted by the highest Z score down. Defaults to True.
- **MAD** (*bool*) – calculates the Mean Absolute Difference between the found and the expected distributions; defaults to False.
- **confidence** (*int, float*) – confidence level to draw lower and upper limits when plotting and to limit the top deviations to show. Defaults to None.
- **high_Z** (*int*) – chooses which Z scores to be used when displaying results, according to the confidence level chosen. Defaluts to 'pos', which will highlight only values higher than the expexted frequencies; 'all' will highlight both extremes (positive and negative); and an integer, which will use the first n entries, positive and negative, regardless of whether Z is higher than the confidence or not.
- **limit_N** (*int*) – sets a limit to N as the sample size for the calculation of the Z scores if the sample is too big. Defaults to None.
- **MSE** (*bool*) – calculates the Mean Square Error of the sample; defaults to False.
- **chi_square** – calculates the chi_square statistic of the sample and compares it with a critical value, according to the confidence level chosen and the series's degrees of freedom. Defaults to False. Requires confidence != None.
- **KS** – calculates the Kolmogorov-Smirnov test, comparing the cumulative distribution of the sample with the Benford's, according to the confidence level chosen. Defaults to False. Requires confidence != None.
- **show_plot** (*bool*) – draws the test plot.

- **save_plot** (*str*) – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** (*dict*) – any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

Returns

DataFrame of the test chosen, but applied on Second Order pre- processed data.

1.1.2 benford.expected module

class benford.expected.**First**(*digs*, *plot=True*, *save_plot=None*, *save_plot_kwargs=None*)

Bases: pandas.core.frame.DataFrame

Holds the expected probabilities of the First, First Two, or First Three digits according to Benford's distribution.

Parameters

- **digs** – 1, 2 or 3 - tells which of the first digits to consider: 1 for the First Digit, 2 for the First Two Digits and 3 for the First Three Digits.
- **plot** – option to plot a bar chart of the Expected proportions. Defaults to True.
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

class benford.expected.**Second**(*plot=True*, *save_plot=None*, *save_plot_kwargs=None*)

Bases: pandas.core.frame.DataFrame

Holds the expected probabilities of the Second Digits according to Benford's distribution.

Parameters

- **plot** – option to plot a bar chart of the Expected proportions. Defaults to True.
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

class benford.expected.**LastTwo**(*num=False*, *plot=True*, *save_plot=None*, *save_plot_kwargs=None*)

Bases: pandas.core.frame.DataFrame

Holds the expected probabilities of the Last Two Digits according to Benford's distribution.

Parameters

- **plot** – option to plot a bar chart of the Expected proportions. Defaults to True.

- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension. Only available when plot=True.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html Only available when plot=True and save_plot is a string with the figure file path/name.

1.1.3 benford.stats module

benford.stats.**Z_score**(*frame*, *N*)

Computes the Z statistics for the proportions studied

Parameters

- **frame** – DataFrame with the expected proportions and the already calculated Absolute Diferences between the found and expeccted proportions
- **N** – sample size

Returns Series of computed Z scores

benford.stats.**chi_sq**(*frame*, *ddf*, *confidence*, *verbose=True*)

Comnputes the chi-square statistic of the found distributions and compares it with the critical chi-square of such a sample, according to the confidence level chosen and the degrees of freedom - len(sample) -1.

Parameters

- **frame** – DataFrame with Found, Expected and their difference columns.
- **ddf** – Degrees of freedom to consider.
- **confidence** – Confidence level to look up critical value.
- **verbose** – prints the chi-squre result and compares to the critical chi-square for the sample. Defaults to True.

Returns

The computed Chi square statistic and the critical chi square (according) to the degrees of freedom and confidence level, for comparison. None if confidence is None

benford.stats.**chi_sq_2**(*frame*)

Computes the chi-square statistic of the found distributions

Parameters **frame** – DataFrame with Found, Expected and their difference columns.

Returns The computed Chi square statistic

benford.stats.**kolmogorov_smirnov**(*frame*, *confidence*, *N*, *verbose=True*)

Computes the Kolmogorov-Smirnov test of the found distributions and compares it with the critical chi-square of such a sample, according to the confidence level chosen.

Parameters

- **frame** – DataFrame with Foud and Expected distributions.
- **confidence** – Confidence level to look up critical value.
- **N** – Sample size
- **verbose** – prints the KS result and the critical value for the sample. Defaults to True.

Returns

The Suprem, which is the greatest absolute difference between the Found and the expected proportions, and the Kolmogorov-Smirnov critical value according to the confidence level, for ccomparison

benford.stats.**kolmogorov_smirnov_2**(*frame*)
Computes the Kolmogorov-Smirnov test of the found distributions

Parameters **frame** – DataFrame with Foud and Expected distributions.

Returns

The Suprem, which is the greatest absolute difference between the Found end th expected proportions

benford.stats.**mad**(*frame*, *test*, *verbose=True*)
Computes the Mean Absolute Deviation (MAD) between the found and the expected proportions.

Parameters

- **frame** – DataFrame with the Absolute Deviations already calculated.
- **test** – Test to compute the MAD from (F1D, SD, F2D...)
- **verbose** – prints the MAD result and compares to limit values of conformity. Defaults to True.

Returns

The Mean of the Absolute Deviations between the found and expected proportions.

benford.stats.**mse**(*frame*, *verbose=True*)
Computes the test's Mean Square Error

Parameters

- **frame** – DataFrame with the already computed Absolute Deviations between the found and expected proportions
- **verbose** – Prints the MSE. Defaults to True.

Returns Mean of the squared differences between the found and the expected proportions.

1.1.4 benford.viz module

benford.viz.**plot_expected**(*df*, *digs*, *save_plot=None*, *save_plot_kwargs=None*)
Plots the Expected Benford Distributions

Parameters

- **df** – DataFrame with the Expected Proportions
- **digs** – Test's digit
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html

benford.viz.**plot_digs**(*df*, *x*, *y_Exp*, *y_Found*, *N*, *figsize*, *conf_Z*, *text_x=False*, *save_plot=None*, *save_plot_kwargs=None*)
Plots the digits tests results

Parameters

- **df** – DataFrame with the data to be plotted
- **x** – sequence to be used in the x axis
- **y_Exp** – sequence of the expected proportions to be used in the y axis (line)
- **y_Found** – sequence of the found proportions to be used in the y axis (bars)
- **N** – lenght of sequence, to be used when plotting the confidence levels
- **figsize** – tuple to state the size of the plot figure
- **conf_Z** – Confidence level
- **save_pic** – file path to save figure
- **text_x** – Forces to show all x ticks labels. Defaluts to True.
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html

benford.viz.**plot_sum**(*df*, *figsize*, *li*, *text_x=False*, *save_plot=None*, *save_plot_kwargs=None*)

Plots the summation test results

Parameters

- **df** – DataFrame with the data to be plotted
- **figsize** – sets the dimensions of the plot figure
- **li** – value with which to draw the horizontal line
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html

benford.viz.**plot_ordered_mantissas**(*col*, *figsize=(12, 12)*, *save_plot=None*, *save_plot_kwargs=None*)

Plots the ordered mantissas and compares them to the expected, straight line that should be formed in a Benford-cmpliant set.

Parameters

- **col** (*Series*) – column of mantissas to plot.
- **figsize** (*tuple*) – sets the dimensions of the plot figure.
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html

benford.viz.**plot_mantissa_arc_test**(*df*, *gravity_center*, *grid=True*, *figsize=12*, *save_plot=None*, *save_plot_kwargs=None*)

Draws thee Mantissa Arc Test after computing X and Y circular coordinates for every mantissa and the center of gravity for the set

Parameters

- **df** (*DataFrame*) – pandas DataFrame with the mantissas and the X and Y coordinates.

- **gravity_center** (*tuple*) – coordinates for plottling the gravity center
- **grid** (*bool*) – show grid. Defaults to True.
- **figsize** (*int*) – figure dimensions. No need to be a tuple, since the figure is a square.
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html

benford.viz.**plot_roll_mse**(*roll_series*, *figsize*, *save_plot=None*, *save_plot_kwargs=None*)
Shows the rolling MSE plot

Parameters

- **roll_series** – pd.Series resultant form rolling mse.
- **figsize** – the figure dimensions.
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html

benford.viz.**plot_roll_mad**(*roll_mad*, *figsize*, *save_plot=None*, *save_plot_kwargs=None*)
Shows the rolling MAD plot

Parameters

- **roll_mad** – pd.Series resultant form rolling mad.
- **figsize** – the figure dimensions.
- **save_plot** – string with the path/name of the file in which the generated plot will be saved. Uses matplotlib.pyplot.savefig(). File format is infered by the file name extension.
- **save_plot_kwargs** – dict with any of the kwargs accepted by matplotlib.pyplot.savefig() https://matplotlib.org/api/_as_gen/matplotlib.pyplot.savefig.html

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

2.1 On GitHub

2.1.1 Package

2.1.2 Demo Jupyter Notebook

Python Module Index

b

Index

R

S

T

U

V

Z