Documentation

Nathan Hughes

March 7, 2018

Contents

1 ct_data.py

1.1 CTData

1.1.1 get_data

Returns the dataframe used in this class

1.1.2 gather_data

this function gathers together all the data of interest @param folder is a starting folder @returns tuple of (seed files, rachis files)

1.1.3 make_dataframe

this function returns a dataframe of grain parameters and optionally of the rachis top and bottom @param grain_files is the output from gather_data @param rachis_files is an optional output from gather_data also @returns a dataframe of the information pre-joining

1.1.4 clean_data

Following parameters outlined in the CT software documentation I remove outliers which are known to be errors

1.1.5 get_files

Returns a tuple of grain files and rachis files

1.1.6 fix_colnames

Because Biologists like to give data which are not normalised to any degree this function exists to attempt to correct the grouping columns, after standardisation https://github.com/SirSharpest/CT_Analysing_Library/issues/2 this shouldn't be needed anymore, but kept for legacy issues that could arise!

1.1.7 join_spikes_by_rachis

So important part of this function is that we accept that the data is what it is that is to say: rtop, rbot and Z are all orientated in the proper direction

It's main purpose is to join split spikes by rachis nodes identified in the analysis process

@param grain_df is the grain dataframe to take on-board

1.1.8 remove_percentile

This function is targeted at removing a percentile of a dataframe it uses a column to decide which to measure against. By default this will remove everything above the percentile value

@param df is the dataframe to manipulate @param column is the attribute column to base the removal of @param target_percent is the percentage to aim for @param bool_below is a default param which if set to True will remove values below rather than above percentage

1.1.9 get_spike_info

This function should do something akin to adding additional information to the data frame

@note there is some confusion in the NPPC about whether to use folder name or file name as the unique id when this is made into end-user software, a toggle should be added to allow this

1.1.10 aggregate_spike_averages

This will aggregate features (specified by attributes) into their medians on a per-spike basis.

Makes direct changes to the dataframe (self.df)

@param attributes list of features to average

1.1.11 find_troublesome_spikes

This will attempt to identify spikes which are not performing as expected

The default criteria for this is simply a count check So it requires that aggregate_spike_averages has been run

@returns a dataframe with candidates for manual investigation

1.1.12 make_plot

Returns false if plot could not be created for invalid parameters

2 data_transforms.py

2.0.1 standarise_data

This is to conform with the likes of PCA, the following text is borrowed from: https://towardsdatascience.com/pca-using-python-scikit-learn-e653f8989e60 Some code also also heavily borrowed from this page and I take minimal credit for it.

PCA is effected by scale so you need to scale the features in your data before applying PCA. Use StandardScaler to help you standardize the datasets features onto unit scale (mean = 0 and variance = 1) which is a requirement for the optimal performance of many machine learning algorithms. If you want to see the negative effect not scaling your data can have, scikit-learn has a section on the effects of not standardizing your data.

uses: $\frac{x_i mean(x)}{stdev(x)} assumes Normal Distribution$

2.0.2 perform_pca

This function will perform a PCA and return the principle components as a dataframe.

@param n_components components to check form @param df dataframe of the data to analyse @param features features from the dataframe to use @param groupby the column in the df to use @param standardise=False asks whether to standardise the data prior to PCA

3 graphing.py

3.1 Error

Base class for other exceptions

3.2 InvalidPlot

Except to trigger when a graph is given wrong args

3.2.1 percentile_grid

3.2.2 qq_grid

3.2.3 plot_boxplot

This should just create a single boxplot and return the figure and an axis, useful for rapid generation of single plots Rather than the madness of the plural function

3.2.4 qqplot

What's a QQ plot? https://stats.stackexchange.com/questions/139708/qq-plot-in-python

3.2.5 plot_boxplots

3.2.6 plot_histogram

Simple histogram function

returns a plot axes

3.2.7 check_var_args

Helper function to fix bad arguments before they get used in evaluations

4 statistical_tests.py

4.0.1 test_normality

https://stackoverflow.com/a/12839537

Null Hypothesis is that X came from a normal distribution

which means: If the p-val is very small, it means it is unlikely that the data came from a normal distribution

As for chi-square: https://biology.stackexchange.com/questions/13486/deciding-between-chi-square-and-t-tes