sedimentanalyst

Release 0.0.1

Beatriz Negreiros & Federica Scolari

Mar 06, 2022

CONTENTS

Sediment Analyst is a modularized Python package and dash app that enables sedimentological analyses. By using sieving datasets as input, Sediment Analyst computes sediment statistics. For a complete list of computed statistics see section below (*Outputs*). For using our app, please watch our video tutorial here.

Important: Checkout the package requirements file in the Github repository for installing the necessary processing libraries.

CHAPTER

ONE

OUTPUTS AND CAPABILITIES

Sediment Analyst computes the following:

- **A summary of sediment characteristics, which can be exported as csv:**
  - d10, d16, d25, d30, d50, d84, d90.
  - Mean grain size, geometric mean grain size (Bunte and Abt, 2001), grain size standard deviation, geometric standard deviation (Frings et al., 2011).
  - Sorting index, Fredle index.
  - Skewness and kurtosis.
  - Coefficient of uniformity, curvature coefficient.
  - **Porosity estimators according to empirical equations available in the literature:**
    - Carling and Reader (1982)
    - Wu and Wang (2006)
    - Wooster et al. (2008): recommended for gravel-beds with geometric standard deviation between 0.004 m and 0.018 m.
    - Frings et al. (2011)
  - Hydraulic conductivity estimators computed with the Kozeny-Carman Equation. Hydraulic Conductivity (kf) is computed in m/s with each of the above-mentioned computed porosity values.
  - Cumulative percentages according to the Wentworth scale.
- **Cumulative grain size distribution curves, which are available as:**
  - Static plots per sample with the *analyzer* subpackage.
  - Interactive plots with user-selected samples using the *app* subpackage.

1.1 sedimentanalyst

1.1.1 sedimentanalyst package

Subpackages

sedimentanalyst.analyzer package

Submodules

sedimentanalyst.analyzer.config module

This module contains all the imported packages (dependencies) and user inputs for running the classes StatisticalAnalyzer and StaticPlotter.

sedimentanalyst.analyzer.config.**get_input**()

sedimentanalyst.analyzer.main module

Main script to exemplify the use of Sediment Analyst

Authors: Beatriz Negreiros and Federica Scolari

sedimentanalyst.analyzer.main.**main**()

sedimentanalyst.analyzer.static_plotter module

Module designated for the class StaticPlotter

Author : Federica Scolari

class sedimentanalyst.analyzer.static_plotter.**StaticPlotter**(*analyzer*)

Bases: object

A class for creating the static plots of the cumulative grain size distribution curves

analyzer
StatisticalAnalyzer object containing all the computed sample statistics. For more information check the Class StatisticalAnalyzer

Type *StatisticalAnalyzer*

cum_df
DataFrame containing the Grain Sizes [mm] and the corresponding Cumulative Percentage [%]

cum_plotter(*output*)
Plots the cumulative grain size distribution curve for each sample

cum_plotter(*output*)
Method to output the cumulative grain size distribution curve and save it as an image.

Parameters **output** (*str*) – Name of the image containing the outputs

Returns None

sedimentanalyst.analyzer.statistical_analyzer module

Module designated for class StatisticalAnalyzer

Author : Beatriz Negreiros

class sedimentanalyst.analyzer.statistical_analyzer.**StatisticalAnalyzer**(*sieving_df=None*, *metadata=None*)

Bases: object

A class for computing statistical sedimentological parameters using sieving datasets (class weights and grain size).

original_df
dataframe containing in the first column the grain sizes diameters (in mm) and in the second column the fraction mass that passes through the corresponding diameter.

Type df

cumulative_df
dataframe containing in the first column the grain sizes diameters (in mm) and in the second column the cumulative percentages (% in mass, in grams) that passes through the corresponding grain size diameters.

Type df

statistics_df
dataframe containing all the statistics of the sample, which includes: d10, d16, d25, d30, d50, d60, d75, d84, d90, Mean Grain Site dm [mm], Geometrical mean grain size dg [mm], Sorting Index, Fredle Index, Grain Size standard deviation, skewness, kurtosis, coefficient of uniformity Cu, curvature coefficient Cc.

Type df

porosity_conductivity_df
dataframe containing the porosity estimators (estimated from the grain size analysis) according to different literature, as well as the corresponding hydraulic conductivity estimator for each of the porosity values according to the Kozeny Carman Equation.

Type df

samplename
sample name

Type str

coords
x and y coordinates, in this order

Type tuple

porosity
porosity values set up by the user, possibly via alternative measurements, such as with photogramic approaches.

Type float

sf_porosity
sphericity index. For rounded sediments it equals 6.10

Type float

compute_cumulative_df(*df*)
computes cumulative_df dataframe

compute_statistics_df(*df*)
computes statistics_df dataframe

compute_porosity_conductivity_df(*df*)
computes porosity_conductivity_df dataframe

Note: See more on the determination of riverbed porosity from Freezecore samples via a Structure from Motion approach at Seitz 2020.

---

compute_cumulative_df()
Compute two new columns in the grain size dataframe, which are + Percentage Fraction [%] and + Cumulative Percentage [%]

compute_porosity_conductivity_df()
Compute porosity predictors and corresponding hydraulic conductivities (for each estimated porosity value)

compute_statistics_df()
Fills a dataframe (self.statistics_df) with all relevant statistics by calling smaller private methods

print_excel(*file_name='statistics.xlsx'*)
Print all attribute dataframes into excel sheet output is saved into local folder "outputs"

Parameters **file_name** (*str*) – Path to save the file

sedimentanalyst.analyzer.utils module

Module containing auxiliary functions to handle the StatisticalAnalyzer class and for running main.py

Author: Beatriz Negreiros and Federica Scolari

sedimentanalyst.analyzer.utils.**append_global**(*obj=None*, *df=None*)
A function to append all information stemming from the class Statistical Analyzer into one dataframe for further filtering and analyses

Parameters

- **obj** (StatisticalAnalyzer) –
- **df** (*df*) –

Returns appended dataframe with statistics of sample file

Return type df

sedimentanalyst.analyzer.utils.**extract_df**(*dic=<built-in function input>*, *file=None*)
Function to extract parsed datafiles and tabularize it into dataframe.

Parameters

- **dic** (*dict*) – global input parameters that can be altered in the config.py file
- **file** (*str*) – path name of the file containing a sieving sample

Returns

dataframe containing grain sizes and class weights (parsed according to the config.py) list: list of sample's information as following: [samplename, sampledate, (lat, long), porosity,

sf_porosity], parsed accoridng to the config.py.

Return type df

sedimentanalyst.analyzer.utils.**find_files**(*folder=None*)
Lists the files in the folder indicated

Parameters **folder** (*str*) – path of the folder to scan (to look for .xlxs files)

Returns list of strings from addresses of all files inside the folder

Return type list

Module contents

sedimentanalyst.app package

Submodules

sedimentanalyst.app.accessories module

Accessory elements for the web app

Author: Beatriz Negreiros

class sedimentanalyst.app.accessories.**Accessories**

Bases: object

A class for allocating accessories elements for the Dash app, including layout and Dash component settings, extensive callouts and parsing of input contents.

style_upload
style information for a dropbox component

Type dict

intro_text
markdown test for introducing the app

Type dash.dcc.Markdown.Markdown

inputs_text
markdown text for explaining the inputs

Type dash.dcc.Markdown.Markdown

img_style
style information for formatting images

Type dict

input_boxes
list of Input objects for enabling user to enter the indexing information to read from user's files

Type list

style_graph

Type dict

style_statistic
style information for the statistic dropdown

Type dict

parse_contents(*tuple*)
tuple of object (sedimentanalyst.analyzer.StatisticalAnalyzer) plus an object of

type html.Div with reading messages.

parse_contents(*contents*, *filename*, *date*, *input_dict_app*)

Parameters

- **contents** (*dash.dcc.Input.Input*) – Contents of the file containing the sample data (class weights and
- **sizes)** (*corresponding grain*) –
- **filename** (*dash.dcc.State.State*) – Filename
- **date** (*dash.dcc.State.State*) – date of last modified
- **input_dict_app** (*dict*) – Index parameters input by the user necessary to read and parse the contents of the file

Returns object for accessing necessary attributes of the class.

Return type *StatisticalAnalyzer*

sedimentanalyst.app.appconfig module

sedimentanalyst.app.interac_plotter module

Module designated for the class InteractivePlotter

Author : Federica Scolari

class sedimentanalyst.app.interac_plotter.**InteractivePlotter**(*df*)

Bases: object

A class for creating interactive plots for the comparison of the statistical analysis results

df

DataFrame containing the information from the statistical analysis

Type pandas.core.frame.DataFrame

convert_coordinates(*df*, *projection*)

Transforms the coordinates of a given projection to degrees

create_map(*df*, *projection*, *samples=None*)

Creates a scatter map

plot_barchart(*param*, *samples*)

Plots the user-selected parameter for all samples in a bar chart

plot_gsd(*samples*)

Plots the cumulative grain size distribution curve for all samples using a line chart

plot_diameters(*samples*)

Plots the calculated sediment diameters in a bar chart for all samples

convert_coordinates(*df*, *projection*)

Method which transforms the coordinates of a give projection to degrees.

Parameters

- **df** (*pandas.core.frame.DataFrame*) – DataFrame on which the coordinate transformation is applied
- **projection** (*str*) – Name of the initial projection

Returns DataFrame object on which the coordinate transformation has been applied

Return type pandas.core.frame.DataFrame

create_map(*df*, *projection='epsg:3857'*, *samples=None*)
Creates a scatter map based on the DataFrame.

Parameters

- **df** (*DataFrame*) – DataFrame on which the coordinate transformation has been applied
- **projection** (*str*) – Name of the initial projection
- **samples** (*list*) – Names of the collected samples

Returns Figure object that allows the visualization of the Open Street map of the area where the samples were collected

Return type plotly.graph_objects.Figure

plot_barchart(*param*, *samples*)
Method that outputs the results in a bar chart for the interactive comparison of the results.

Parameters

- **param** (*str*) – Statistical parameters selectable from the user
- **samples** (*list*) – Names of the collected samples

Returns

Figure object that allows the visualization of the plot of a bar chart in which the statistical parameter selected by the user is shown for each selected sample

Return type plotly.graph_objects.Figure

plot_diameters(*samples*)
Method which plots the calculated sediment diameters in a bar chart for all selected samples.

Parameters **samples** (*list*) – Names of the collected samples

Returns Figure object allowing to visualize the calculated diameters (d10, d16, d25, d30, d50, d60, d75, d84 and d90) for all the collected samples

Return type plotly.graph_objects.Figure

plot_gsd(*samples*)
Method which plots the cumulative grain size distribution curve for all selected samples.

Parameters **samples** (*list*) – Names of the collected samples

Returns

Figure object enabling the visualization of the plot of the grain size distribution curve for all collected sample by using a line chart

Return type plotly.graph_objects.Figure

sedimentanalyst.app.web_application module

Stand-alone designated for the web application

Author : Beatriz Negreiros

sedimentanalyst.app.web_application.**download_summary_stats**(*data*, *n_clicks*)

sedimentanalyst.app.web_application.**parse_and_analyse**(*list_of_contents*, *list_of_names*, *list_of_dates*, *input_dict_in_layout*, *click*)

sedimentanalyst.app.web_application.**save_inputs**(*header*, *gs_clm*, *cw_clm*, *n_rows*, *porosity*, *sf_porosity*, *index_lat*, *index_lon*, *sample_name_index*, *sample_date_index*, *projection*, *n_clicks*)

sedimentanalyst.app.web_application.**update_barchart**(*data*, *stat_value*, *samples*)

sedimentanalyst.app.web_application.**update_diameters**(*data*, *samples*)

sedimentanalyst.app.web_application.**update_gsd**(*data*, *samples*)

sedimentanalyst.app.web_application.**update_map**(*data*, *dict_to_get_proj*, *samples*)

sedimentanalyst.app.web_application.**update_sample_id**(*n_clicks*, *data*)

sedimentanalyst.app.web_application.**update_stat_drop**(*n_clicks*, *data*)

Module contents

Module contents

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

S

INDEX

O

P

S

U