data_profiling Documentation

Release Beta

Adriano Cosi

Sep 02, 2019

CONTENTS:

class `src.DataProfiler.`**`DataProfiler`**

Class for data profiling

`convert_column`(*dataframe*, *column*, *convert_to='datetime'*)

Parameters

- **`dataframe`** – Dataframe with column to convert
- **`column`** – Column to convert
- **`convert_to`** – One of "datetime" and "categorical". Specify column type to convert data to. Defaults to 'datetime'

Returns Converted column

Note Works only on 'object' (i.e. generic) type dataframe columns.

`describe_columns`(*dataframe, columns, percentiles=[0.25, 0.5, 0.75], identify_outliers=True, exclude_outliers_from_graph=True, show_graphs=False*)

Return a dictionary of column name to associated summary statistics

Parameters

- **`dataframe`** – DataFrame containing column(s) to describe
- **`columns`** – Column(s) to describe in dataframe
- **`percentiles`** – Percentiles (range 0 to 1) to calculate for summary statistics. This has effect only on numerical columns
- **`identify_outliers`** – If true finds outliers in each column to plot if numerical
- **`exclude_outliers_from_graph`** – If true it excludes the detected outliers from the graph

`descriptive_res`

alias of `Columns_Summary`

`distribution_over_time_for_columns`(*dataframe*, *columns_to_plot*, *date_column*, *start_date=None*, *end_date=None*, *frequency='day'*, *frequency_multiplier=1*)

Parameters

- **`dataframe`** – Dataframe to plot
- **`columns_to_plot`** – List of columns to plot
- **`date_column`** – The date column (does not need to be converted to date type)
- **`start_date`** – Start plotting from this date
- **`end_date`** – Limit plot to this date
- **`frequency`** – One of: day,week,month,year. Frequency at which the data is plotted
- **`frequency_multiplier`** – Integer, modifies frequency by that integer (e.g. if frequency=day and multiplier=2 final frequency is 2 days)

Returns Dictionary of column plotted to matplotlib plot

`identify_outliers`(*dataframe*, *column*, *column_stats=Empty DataFrame Columns: [] Index: []*)

Parameters

- **`dataframe`** – Input dataframe
- **`column`** – Input column

- **column_stats** – If passed avoids calculating summary stats for data.

Used to find outliers :return: boolean array for indexing outlier values

plot_descriptive_graphs_for_column(*dataframe*, *column*, *outliers_ind=None*, *show=False*)

Parameters

- **dataframe** – Input dataframe
- **column** – Column to plot
- **outliers_ind** – Boolean array for indexing outliers
- **show** – If to show graphs when running the code

Returns Mapping of columns plotted to graphs types

prepare_dataframe(*dataframe*, *date_columns=[]*, *categorical_columns=[]*)

Parameters

- **dataframe** – Dataframe with columns to convert
- **date_columns** – List of date columns in dataframe
- **categorical_columns** – List of categorical columns in dataframe

Returns Dataframe with converted columns

summary_stats(*dataframe, column, percentiles=[0.25, 0.5, 0.75], print_summary=True*)

Parameters

- **dataframe** – Input data
- **column** – Column to describe
- **percentiles** – Percentiles for numerical column (range 0 to 1)

Returns Summary statistics for column

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX