`Portfolio` Class Cheat Sheet

ⓜ `Portfolio(self, name, spec_list, uw=None)`

The `Portfolio` call signature requires a name and `spec_list` a DecL program, a list of `Aggregate` objects or kwargs, or names known to the `Underwriter`, or a multivariate loss sample in a pandas DataFrame. The following tables show all **ⓜ** `methods`, **ⓢ** `static methods`, and fields or properties (used interchangeably). Comments elucidate the meaning of more obscure entries. Internal methods and fields are not shown.

1. Specification & creation

`name` *more*
`spec_list` *more*
`uw` *more*
`n_units`, `agg_list` (list of `Aggregate` objects),
`line_names`, `line_names_ex`, `unit_names` (unit ← line), `unit_names_ex`, `line_name_pipe`, `program` (DecL program), `pprogram` (pretty printed), `spec` (constructor kwarg dictionary; `Aggregate(**spec)` re-creates the object), `spec_ex` (adds meta information),
ⓜ `nice_program`, **ⓢ** `from_DataFrame`,
ⓢ `from_dict_of_aggs`, **ⓢ** `from_Excel`,
ⓢ `create_from_sample`

2. Update

`log2`, `bs`, `sev_calc` (discrete=round, forward, backwards), `discretization_calc` (distribution, survival, both), `normalize`, `padding`, `tilt_amount`, `approx_freq_ge`, `approx_type` (exact, slognorm, sgamma), **ⓜ** `best_bucket`, **ⓜ** `recommend_bucket`,
ⓜ `update`, **ⓜ** `add_exa`, **ⓜ** `add_exa_details`,
ⓜ `add_exa_sample`, **ⓜ** `trim_df`, **ⓜ** `ft` & **ⓜ** `ift`
(FFT and inverse FFT), **ⓜ** `remove_fuzz`, **ⓜ** `set_a_p`

3. Moments

est prefix=estimated from FFT approximation
`agg_m`, `agg_cv`, `agg_sd`, `agg_var`, `agg_skew`,
`est_m`, `est_cv`, `est_sd`, `est_var`, `est_skew`,
`ex`

4. Statistical functions

ⓜ `cdf`, **ⓜ** `sf` (survival), **ⓜ** `pdf`, **ⓜ** `pmf`, **ⓜ** `q` (lower quantile=VaR), **ⓜ** `tvar`, **ⓜ** `tvar_threshold`, **ⓜ** `var`,
ⓜ `var_dict`, **ⓜ** `density_sample`, **ⓜ** `percentiles`,
ⓜ `sample`, **ⓜ** `sample(_density)_compare`,

5. Validation

`describe` (validation statistics),
`valid` (true=all components and total "not unreasonable" or false),
`validation_eps` (validation epsilon threshold 1e-04),
ⓜ `audits`, **ⓜ** `uat`, **ⓜ** `uat_differential`,
ⓜ `uat_interpolation_functions`

6. Output dataframes

`density_df`[1] (main output),
`report_df` (component, mixture & empirical stats),
`statistics`, `statistics_df`, `audit_df`,
`augmented_df`, `independent_audit_df`,
`independent_density_df`, `priority_analysis_df`,
ⓜ `make_audit_df`, **ⓜ** `make_all`, **ⓜ** `report`

7. Reinsurance

None – applies at the component level

8. Visualization & exhibits

ⓜ `plot`, **ⓜ** `scatter`, **ⓜ** `twelve_plot`,
ⓜ `biv_contour_plot`,
ⓜ `analyze_distortion_plots`,
ⓜ `natural_profit_segment_plot`,
ⓜ `profit_segment_plot`, `figure` (return last figure),
ⓜ `limits`, `line_renamer`,
`premium_capital_renamer`, `renamer`,
ⓜ `short_renamer`, `stat_renamer`, `tm_renamer`,
ⓜ `show_enhanced_exhibits`,
`EX_accounting_economic_balance_sheet`,
`EX_multi_premium_capital`, `EX_premium_capital`

9. Risk and pricing

ⓜ `accounting_economic_balance_sheet`,
ⓜ `analysis_collateral`, **ⓜ** `analysis_priority`,
ⓜ `analyze_distortion(s|_add_comps)`,
ⓜ `apply_distortion(s)`, `assets_2_epd`,
ⓜ `bodoff`, **ⓜ** `calibrate_blends`,
ⓜ `calibrate_distortion(s)`, **ⓜ** `cotvar`,
`dist_ans`, `distortion`, `distortion_df`, `dists`,
`epd_2_assets`, **ⓜ** `equal_risk_epd`,
ⓜ `equal_risk_var_tvar`, **ⓜ** `gamma`, **ⓜ** `gradient`,
ⓜ `merton_perold`, **ⓜ** `multi_premium_capital`,
ⓜ `premium_capital`, **ⓜ** `price`, **ⓜ** `price_ccoc`,
ⓜ `pricing_bounds`, `priority_capital_df`,
ⓜ `stand_alone_pricing`,

10. Approximations

ⓜ `approximate`, **ⓜ** `as_severity`, **ⓜ** `collapse`

11. Meta

`audit_percentiles`, `hash_rep_at_last_update`,
`info` (text meta info), **ⓜ** `json` (persist to json),
`last_update`, **ⓜ** `more(regex)` (print all methods and fields matching regex), **ⓜ** `save`, **ⓜ** `snap`[1] (snap argument to index)

Notes:

[1]: matches `Aggregate`

Any vectorizable input accepts numeric or iterable datatypes.

Abbreviations: gcn=gross (subject), ceded, and net; stats: m=mean, cv=coefficient of variation, sd=standard deviation, var=variance, skew(ness); VaR=value-at-risk

`aggregate v.0.20.0`
Confidence and precision meets
ease of use in actuarial analysis
© Stephen J Mildenhall
2023-07-22 18:13:56+01:00