Package 'riskweightedassets'

September 24, 2026

Type Package

Title Reproducible Risk-Weighted Asset Calculations

Version 1.1.1

Description Provides transparent, deterministic and auditable calculations of risk-weighted assets, own-funds requirements, interest-rate risk in the banking book and related capital metrics. It supports canonical in-memory tables and versioned spreadsheet datasets, strict validation, synthetic reference profiles, bitemporal snapshots, calculation controls and traceable regulatory source metadata. Methods are parameterised against the European Parliament and Council (2013) Capital Requirements Regulation <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32013R0575> and its amending Regulation (EU) 2024/1623 <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1623>. A granular analyst API exposes individual formulae, domain views, controls, schemas and auditable parameter overrides. The implementation is intended for analytical, educational and model-validation use and does not constitute legal or supervisory advice.

License GPL-3

Copyright 2026 RiskDataScience GmbH

URL https://github.com/rds0001/risk-weighted-assets-r

BugReports https://github.com/rds0001/risk-weighted-assets-r/issues

Encoding UTF-8

Date 2026-09-14

Depends R (>= 4.1.0)

Imports digest, jsonlite, openxlsx, readxl, utils, yaml

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.3.1

VignetteBuilder knitr

NeedsCompilation no

Author Dimitrios Geromichalos [cre],
RiskDataScience GmbH [aut, cph]

Maintainer Dimitrios Geromichalos <riskdatascience@web.de>

Repository CRAN

Date/Publication 2026-09-24 13:50:10 UTC

Contents

`analyze_capital_adequacy` *Analyse Capital Adequacy, Leverage and MREL/TLAC*

Description

Analyse Capital Adequacy, Leverage and MREL/TLAC

Usage

```
analyze_capital_adequacy(x, view = c("applied", "fully_loaded"))
```

Arguments

`x`	Canonical tables or an existing calculation result.
`view`	`"applied"` or `"fully_loaded"`.

Value

An `rwa_domain_analysis`.

analyze_counterparty_risk *Analyse Counterparty Credit Risk, CVA and Settlement Risk*

Description

Analyse Counterparty Credit Risk, CVA and Settlement Risk

Usage

```
analyze_counterparty_risk(x, view = c("applied", "fully_loaded"))
```

Arguments

x	Canonical tables or an existing calculation result.
view	"applied" or "fully_loaded".

Value

An rwa_domain_analysis.

analyze_credit_risk *Analyse Credit Risk*

Description

Analyse Credit Risk

Usage

```
analyze_credit_risk(x, view = c("applied", "fully_loaded"))
```

Arguments

x	Canonical tables or an existing calculation result.
view	"applied" or "fully_loaded".

Value

An rwa_domain_analysis with relevant metrics, tables and controls.

See Also

calculate_tables()

Examples

```
analyze_credit_risk(generate_synthetic_tables(bank_profile = "KSA_BANK"))
```

`analyze_icaap` *Analyse ICAAP Economic and Normative Perspectives*

Description

Analyse ICAAP Economic and Normative Perspectives

Usage

```
analyze_icaap(x, view = c("applied", "fully_loaded"))
```

Arguments

`x`	Canonical tables or an existing calculation result.
`view`	`"applied"` or `"fully_loaded"`.

Value

An `rwa_domain_analysis`.

`analyze_irrbb` *Analyse IRRBB and CSRBB*

Description

Analyse IRRBB and CSRBB

Usage

```
analyze_irrbb(x, view = c("applied", "fully_loaded"))
```

Arguments

`x`	Canonical tables or an existing calculation result.
`view`	`"applied"` or `"fully_loaded"`.

Value

An `rwa_domain_analysis`.

analyze_market_risk *Analyse Market Risk and FRTB*

Description

Analyse Market Risk and FRTB

Usage

```
analyze_market_risk(x, view = c("applied", "fully_loaded"))
```

Arguments

x	Canonical tables or an existing calculation result.
view	"applied" or "fully_loaded".

Value

An rwa_domain_analysis.

analyze_operational_risk *Analyse Operational Risk*

Description

Analyse Operational Risk

Usage

```
analyze_operational_risk(x, view = c("applied", "fully_loaded"))
```

Arguments

x	Canonical tables or an existing calculation result.
view	"applied" or "fully_loaded".

Value

An rwa_domain_analysis.

analyze_output_floor *Analyse the Output Floor*

Description

Analyse the Output Floor

Usage

```
analyze_output_floor(x, view = c("applied", "fully_loaded"))
```

Arguments

x	Canonical tables or an existing calculation result.
view	"applied" or "fully_loaded".

Value

An rwa_domain_analysis.

analyze_securitisation *Analyse Securitisation Risk*

Description

Analyse Securitisation Risk

Usage

```
analyze_securitisation(x, view = c("applied", "fully_loaded"))
```

Arguments

x	Canonical tables or an existing calculation result.
view	"applied" or "fully_loaded".

Value

An rwa_domain_analysis.

as.data.frame.rwa_validation_report *Convert a Validation Report to a Data Frame*

Description

Convert a Validation Report to a Data Frame

Usage

```
## S3 method for class 'rwa_validation_report'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	An rwa_validation_report.
row.names	Unused compatibility argument.
optional	Unused compatibility argument.
...	Unused.

Value

A data frame with one row per structured validation message.

available_rule_sets *Available Rule Sets*

Description

Available Rule Sets

Usage

```
available_rule_sets(tables = NULL)
```

Arguments

tables	Optional canonical tables; bundled reference tables are used otherwise.

Value

The rule-set data frame.

See Also

select_rule_set()

Examples

```
available_rule_sets()
```

`business_indicator_component` *Operational Risk, Output Floor, NPE and Tier-2 Formulae*

Description

Operational Risk, Output Floor, NPE and Tier-2 Formulae

Usage

```
business_indicator_component(
  il,
  ie,
  assets,
  dividends,
  oi,
  oe,
  fi,
  fe,
  trading_pnl,
  banking_pnl,
  parameters = NULL
)

applicable_output_floor_factor(as_of, fully_loaded = FALSE, parameters = NULL)

apply_output_floor(
  u_trea,
  s_trea,
  factor,
  optional_cap = FALSE,
  cap_multiplier = 1
)

npe_unsecured_coverage_factor(year, parameters = NULL)

npe_secured_coverage_factor(year, property_security = TRUE, parameters = NULL)

tier2_eligible_amount(
  current_amount,
  first_day_amount,
  maturity,
  as_of,
  parameters = NULL
)
```

Arguments

il, ie, assets, dividends, oi, oe, fi, fe, trading_pnl, banking_pnl	Numeric vectors for the business-indicator components.
parameters	Optional parameter data frame or store.
as_of	Reporting date.
fully_loaded	Whether to return the fully-loaded floor factor.
u_trea	Unfloored TREA.
s_trea	Standardised TREA.
factor	Output-floor factor.
optional_cap	Whether an optional cap applies.
cap_multiplier	Cap multiple of unfloored TREA.
year	NPE vintage year.
property_security	Whether secured by property.
current_amount, first_day_amount	Tier-2 instrument amounts.
maturity	Tier-2 maturity date.

Value

Numeric, named numeric vector, or output-floor result list.

Examples

```
apply_output_floor(100, 180, 0.725, FALSE, 1)
applicable_output_floor_factor(as.Date("2026-12-31"), FALSE)
```

---

calculate_dataset *Calculate a Canonical Workbook Dataset*

Description

Validates a dataset, creates an immutable deterministic run directory and writes six structured output workbooks plus a JSON run manifest.

Usage

```
calculate_dataset(dataset)
```

Arguments

dataset	Dataset directory containing an inputs subdirectory.

Details

Output is written below `dataset/outputs/<run-id>`. The run ID fingerprints the input files, code and package version.

Value

An `rwa_calculation_result` object with output paths.

See Also

`validate_dataset()`, `generate_synthetic_dataset()`

Examples

```
root <- file.path(tempdir(), "rwa-calculation-example")
dataset <- generate_synthetic_dataset(root, bank_profile = "KSA_BANK")
calculate_dataset(dataset)
```

---

calculate_tables *Calculate Canonical In-Memory Tables*

Description

Validates and calculates canonical tables without producing spreadsheet output. A parallel fully-loaded result is calculated from the same snapshot.

Usage

```
calculate_tables(
  tables,
  run_id = NULL,
  project_root = NULL,
  parameter_overrides = NULL,
  override_reason = NULL,
  override_approved_by = NULL
)
```

Arguments

`tables`	Named list of canonical data frames.
`run_id`	Optional caller-defined run identifier.
`project_root`	Optional root used to resolve local source documents.
`parameter_overrides`	Optional data frame of controlled parameter overrides prepared as described in `override_regulatory_parameters()`.
`override_reason`, `override_approved_by`	Required non-empty governance information when `parameter_overrides` is supplied.

Details

The return includes applied and fully-loaded result tables. All calculations use one official bitemporal snapshot. The function does not write files and does not change the global random-number stream.

Value

An `rwa_calculation_result` object.

See Also

`generate_synthetic_tables()`, `calculate_dataset()`

Examples

```
tables <- generate_synthetic_tables(bank_profile = "KSA_BANK")
result <- calculate_tables(tables)
result$metrics[c("RWEA_KSA", "TREA")]
```

---

compare_calculation_views

Compare Applied and Fully-loaded Metrics

Description

Compare Applied and Fully-loaded Metrics

Usage

```
compare_calculation_views(result)
```

Arguments

`result`	An `rwa_calculation_result`.

Value

A data frame with both values and their difference.

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
compare_calculation_views(r)
```

create_workspace *Create a Writable RWA Workspace*

Description

Exports the package's synthetic reference environment to a caller-controlled directory. Existing non-empty directories require explicit `overwrite`.

Usage

```
create_workspace(path, overwrite = FALSE)
```

Arguments

`path`	Destination directory.
`overwrite`	Whether existing files may be replaced.

Value

An `rwa_workspace` object.

Examples

```
workspace <- create_workspace(file.path(tempdir(), "rwa-example"), overwrite = TRUE)
print(workspace)
```

default_workspace *Default Writable Workspace*

Description

Uses `RWA_WORKSPACE` when it is set, otherwise a `rwa-workspace` directory below the current working directory. The function only describes the path; it does not create or modify it.

Usage

```
default_workspace()
```

Value

An `rwa_workspace` object.

Examples

```
default_workspace()
```

failed_controls *Failed Calculation Controls*

Description

Failed Calculation Controls

Usage

```
failed_controls(result, view = c("applied", "fully_loaded"))
```

Arguments

result	An rwa_calculation_result.
view	"applied" or "fully_loaded".

Value

A data frame containing only failed controls.

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
failed_controls(r)
```

formula_catalog *Formula Catalogue*

Description

Formula Catalogue

Usage

```
formula_catalog(tables = NULL)
```

Arguments

tables	Optional canonical tables; bundled reference tables are used otherwise.

Value

The versioned formula-definition data frame.

See Also

regulatory_parameters()

Examples

```
head(formula_catalog())
```

`generate_synthetic_dataset` *Generate a Synthetic Workbook Dataset*

Description

Generate a Synthetic Workbook Dataset

Usage

```
generate_synthetic_dataset(
  root,
  as_of = NULL,
  version = NULL,
  seed = NULL,
  bank_profile = "MID_SIZE_UNIVERSAL",
  config_root = NULL,
  overwrite = FALSE
)
```

Arguments

`root`	Parent directory for dated datasets.
`as_of`	Reporting date.
`version`	Dataset version.
`seed`	Integer lineage seed.
`bank_profile`	Profile identifier.
`config_root`	Optional profile configuration directory.
`overwrite`	Whether an existing dataset may be replaced.

Details

The function writes 16 canonical workbooks plus a dataset manifest. An existing non-empty target requires `overwrite = TRUE`.

Value

The generated dataset directory.

See Also

validate_dataset(), calculate_dataset()

Examples

```
generate_synthetic_dataset(file.path(tempdir(), "rwa-runs"),
                           bank_profile = "KSA_BANK")
```

---

generate_synthetic_tables

Generate Synthetic Canonical Tables

Description

Materializes a complete synthetic banking profile from immutable package fixtures. The seed is recorded for lineage; no business distributions are generated at runtime.

Usage

```
generate_synthetic_tables(
  as_of = NULL,
  seed = NULL,
  bank_profile = "MID_SIZE_UNIVERSAL",
  config_root = NULL
)
```

Arguments

as_of	Reporting date. Defaults to the profile base date.
seed	Integer lineage seed.
bank_profile	Profile identifier.
config_root	Optional directory containing profile YAML files.

Value

A named list of canonical data frames.

See Also

calculate_tables(), generate_synthetic_dataset()

Examples

```
tables <- generate_synthetic_tables(bank_profile = "KSA_BANK")
length(tables)
```

`irb_asset_correlation` *Internal Ratings Based Formulae*

Description

Direct access to IRB asset correlations, maturity adjustment and capital requirement. Probability, LGD and correlation inputs are decimal rates.

Usage

```
irb_asset_correlation(
  pd,
  annual_sales_million = NULL,
  financial_multiplier = FALSE,
  parameters = NULL
)

irb_retail_correlation(pd, subclass, parameters = NULL)

irb_maturity_coefficient(pd, parameters = NULL)

irb_maturity_factor(pd, maturity, parameters = NULL)

irb_capital_requirement(
  pd,
  lgd,
  correlation,
  maturity,
  apply_maturity_adjustment = TRUE,
  defaulted = FALSE,
  elbe = 0,
  parameters = NULL
)
```

Arguments

`pd`	Probability of default.
`annual_sales_million`	Optional annual sales in millions.
`financial_multiplier`	Whether to apply the financial-sector multiplier.
`parameters`	Optional parameter data frame or store.
`subclass`	Retail exposure subclass.
`maturity`	Effective maturity in years.
`lgd`	Loss given default.

`correlation`	Asset correlation.
`apply_maturity_adjustment`	Whether the maturity adjustment applies.
`defaulted`	Whether the exposure is defaulted.
`elbe`	Best estimate of expected loss for a defaulted exposure.

Value

One numeric coefficient or capital-requirement rate.

See Also

`regulatory_parameters()`

Examples

```
irb_asset_correlation(0.01)
irb_capital_requirement(0.01, 0.45, 0.20, 2.5, TRUE, FALSE, 0)
```

`irrbb_scenario_shock` *IRRBB, FRTB and Economic-capital Formulae*

Description

IRRBB, FRTB and Economic-capital Formulae

Usage

```
irrbb_scenario_shock(scenario, t, parallel, short, long, parameters = NULL)

irrbb_shocked_zero_rate(base, shock, t, parameters = NULL)

present_value_discount_factor(continuous_zero_rate, t)

aggregate_correlated_capital(capitals, correlation)

frtb_scenario_correlation(base, scenario, parameters = NULL)

frtb_quadratic_charge(values, correlation, curvature = FALSE)
```

Arguments

`scenario`	IRRBB or FRTB correlation scenario.
`t`	Time in years.
`parallel`, `short`, `long`	Scenario shocks as decimal rates.

parameters	Optional parameter data frame or store.
base	Base continuously compounded zero rate or base correlation.
shock	Rate shock.
continuous_zero_rate	Continuously compounded zero rate.
capitals	Vector of standalone capital amounts.
correlation	Correlation matrix or scalar intra-bucket correlation.
values	Weighted sensitivities.
curvature	Whether curvature aggregation applies.

Value

Numeric shock, rate, discount factor or aggregated capital.

Examples

```
present_value_discount_factor(0.03, 5)
aggregate_correlated_capital(c(10, 20), matrix(c(1, .25, .25, 1), 2))
```

---

list_reference_datasets

List Bundled Reference Datasets

Description

List Bundled Reference Datasets

Usage

```
list_reference_datasets()
```

Value

A data frame with dataset identifiers and versions.

Examples

```
list_reference_datasets()
```

list_reference_profiles *List Bundled Reference Profiles*

Description

Returns metadata for every synthetic reference profile included with the package. No customer or production data is included.

Usage

```
list_reference_profiles()
```

Value

A data frame with profile identifiers, versions, base dates, seeds, default dataset versions and descriptions.

Examples

```
list_reference_profiles()
```

official_snapshot *Select the Official Bitemporal Snapshot*

Description

Select the Official Bitemporal Snapshot

Usage

```
official_snapshot(tables, as_of_date = NULL, knowledge_time = NULL)
```

Arguments

tables	Canonical table list.
as_of_date	Optional business date; defaults to run_config.
knowledge_time	Optional knowledge timestamp; defaults to run_config.

Value

Canonical tables reduced to the effective official records.

Examples

```
t <- generate_synthetic_tables(bank_profile = "KSA_BANK")
s <- official_snapshot(t)
```

override_regulatory_parameters *Apply Auditable Regulatory Parameter Overrides*

Description

Creates a modified copy of canonical tables. Existing parameter rows are replaced only when key and both dimensions identify exactly one row. The original input is not mutated and an audit record is attached.

Usage

```
override_regulatory_parameters(tables, overrides, reason, approved_by)
```

Arguments

tables	Canonical input tables.
overrides	Data frame with parameter_key, parameter_value and optional dimension_1, dimension_2 columns.
reason	Non-empty business rationale.
approved_by	Non-empty approver or governance reference.

Value

A copied table list carrying an rwa_parameter_overrides audit attribute.

See Also

parameter_overrides(), calculate_tables()

Examples

```
t <- generate_synthetic_tables(bank_profile = "KSA_BANK")
o <- data.frame(parameter_key = "RWA_MULTIPLIER", dimension_1 = "PILLAR1",
                dimension_2 = "", parameter_value = 12.5)
t2 <- override_regulatory_parameters(t, o, "Sensitivity", "Model Risk")
parameter_overrides(t2)
```

parameter_overrides *Parameter Override Audit Trail*

Description

Parameter Override Audit Trail

Usage

```
parameter_overrides(x)
```

Arguments

x	Canonical tables or an rwa_calculation_result.

Value

A data frame of old/new values and governance information.

See Also

override_regulatory_parameters()

Examples

```
parameter_overrides(generate_synthetic_tables(bank_profile = "KSA_BANK"))
```

print.rwa_calculation_result *Print a Calculation Result*

Description

Print a Calculation Result

Usage

```
## S3 method for class 'rwa_calculation_result'
print(x, ...)
```

Arguments

x	An rwa_calculation_result.
...	Unused.

Value

x, invisibly.

print.rwa_domain_analysis *Print a Domain Analysis*

Description

Print a Domain Analysis

Usage

```
## S3 method for class 'rwa_domain_analysis'
print(x, ...)
```

Arguments

x	An rwa_domain_analysis.
...	Unused.

Value

x, invisibly.

print.rwa_validation_report *Print a Validation Report*

Description

Print a Validation Report

Usage

```
## S3 method for class 'rwa_validation_report'
print(x, ...)
```

Arguments

x	An rwa_validation_report.
...	Unused.

Value

x, invisibly.

print.rwa_workspace *Print a Workspace*

Description

Print a Workspace

Usage

```
## S3 method for class 'rwa_workspace'
print(x, ...)
```

Arguments

x	An rwa_workspace.
...	Unused.

Value

x, invisibly.

regulatory_parameter *Read One Regulatory Parameter*

Description

Read One Regulatory Parameter

Usage

```
regulatory_parameter(
  key,
  dimension_1 = "",
  dimension_2 = "",
  parameters = NULL
)
```

Arguments

key	Regulatory parameter key.
dimension_1, dimension_2	Optional parameter dimensions.
parameters	A parameter-store object or parameter data frame. Bundled regulatory defaults are used when omitted.

Value

One numeric parameter value.

See Also

regulatory_parameters()

Examples

```
regulatory_parameter("RWA_MULTIPLIER", "PILLAR1")
```

---

regulatory_parameters *Regulatory Parameter Inventory*

Description

Returns the effective parameter table used by granular formulas. Analysts can inspect every weight and coefficient before calculation.

Usage

```
regulatory_parameters(tables = NULL)
```

Arguments

tables	Optional canonical table list. If omitted, bundled defaults are used.

Value

A data frame containing parameter keys, dimensions and values.

See Also

regulatory_parameter(), override_regulatory_parameters()

Examples

```
head(regulatory_parameters())
```

regulatory_sources *Regulatory Source Metadata*

Description

Returns links and archived checksums for official sources used to design the engine. The source documents themselves are deliberately not redistributed.

Usage

```
regulatory_sources()
```

Value

A data frame. The `redistributed` column is always false for the bundled metadata.

Examples

```
regulatory_sources()
```

rwa_controls *Calculation Controls*

Description

Calculation Controls

Usage

```
rwa_controls(result, view = c("applied", "fully_loaded"))
```

Arguments

`result`	An `rwa_calculation_result`.
`view`	`"applied"` or `"fully_loaded"`.

Value

A data frame containing reconciliation and governance controls.

See Also

`failed_controls()`

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
rwa_controls(r)
```

rwa_metric *Read One Calculation Metric*

Description

Read One Calculation Metric

Usage

```
rwa_metric(result, metric, view = c("applied", "fully_loaded"))
```

Arguments

result	An rwa_calculation_result.
metric	Metric identifier.
view	"applied" or "fully_loaded".

Value

One numeric value.

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
rwa_metric(r, "CET1_RATIO")
```

rwa_metrics *Calculation Metrics*

Description

Calculation Metrics

Usage

```
rwa_metrics(result, view = c("applied", "fully_loaded"))
```

Arguments

result	An rwa_calculation_result.
view	"applied" or "fully_loaded".

Value

A tidy data frame with metric, value, unit and view.

See Also

rwa_metric(), compare_calculation_views()

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
rwa_metrics(r)
```

`rwa_result_table` *Read One Result Table*

Description

Read One Result Table

Usage

```
rwa_result_table(result, table, view = c("applied", "fully_loaded"))
```

Arguments

`result`	An `rwa_calculation_result`.
`table`	Result-table name.
`view`	`"applied"` or `"fully_loaded"`.

Value

A result data frame.

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
rwa_result_table(r, "Capital_Stack")
```

rwa_result_tables *Result Tables*

Description

Result Tables

Usage

```
rwa_result_tables(result, view = c("applied", "fully_loaded"))
```

Arguments

result	An rwa_calculation_result.
view	"applied" or "fully_loaded".

Value

A named list of result data frames.

See Also

rwa_result_table(), rwa_table_names()

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
names(rwa_result_tables(r))
```

rwa_summary *Compact Calculation Summary*

Description

Compact Calculation Summary

Usage

```
rwa_summary(result)
```

Arguments

result	An rwa_calculation_result.

Value

A list with identity, core metrics, controls and overrides.

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
rwa_summary(r)
```

`rwa_table_names` *List Result-Table Names*

Description

List Result-Table Names

Usage

```
rwa_table_names(result, view = c("applied", "fully_loaded"))
```

Arguments

`result`	An `rwa_calculation_result`.
`view`	`"applied"` or `"fully_loaded"`.

Value

Character vector of result-table names.

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
rwa_table_names(r)
```

rwa_validation *Calculation Validation Report*

Description

Calculation Validation Report

Usage

```
rwa_validation(result)
```

Arguments

result	An rwa_calculation_result.

Value

The rwa_validation_report attached to the result.

Examples

```
r <- calculate_tables(generate_synthetic_tables(bank_profile = "KSA_BANK"))
rwa_validation(r)
```

sa_ccr_multiplier_value *Counterparty, SFT, Securitisation, CVA and Settlement Formulae*

Description

Granular formula interface for SA-CCR, SFT, securitisation approaches, BA-CVA and settlement risk. Rates are decimal values.

Usage

```
sa_ccr_multiplier_value(V, C, addon, parameters = NULL)

sa_ccr_exposure_value(V, C, addon, alpha, parameters = NULL)

sft_exposure_value(cash_leg, security_value, security_haircut, fx_haircut)

securitisation_irb_pool_capital(
  rwea_pool_irb_ul,
  el_pool_irb,
  pool_ead,
```

```
  parameters = NULL
)

securitisation_sa_pool_capital(rwea_pool_sa, pool_ead, parameters = NULL)

securitisation_ssfa_coefficient(ka, attachment, detachment, p)

securitisation_ssfa_risk_weight(
  pool_k,
  attachment,
  detachment,
  p,
  floor,
  parameters = NULL
)

securitisation_irba_p(
  pool_type,
  senior,
  effective_number,
  pool_k,
  average_lgd,
  tranche_maturity,
  sts,
  parameters = NULL
)

securitisation_erba_risk_weight(
  cqs,
  maturity,
  senior,
  sts,
  attachment,
  detachment,
  parameters = NULL
)

securitisation_risk_weight(
  approach,
  pool_k,
  attachment,
  detachment,
  p,
  sts = FALSE,
  senior = FALSE,
  resecuritisation = FALSE,
  cqs = 0,
  parameters = NULL
```

```
)

cva_basic_approach_capital(items, parameters = NULL)

settlement_risk_factor(days_late, parameters = NULL)
```

Arguments

V, C	Current value and collateral under SA-CCR.
addon	Aggregate SA-CCR add-on.
parameters	Optional parameter data frame or store.
alpha	SA-CCR regulatory alpha.
cash_leg, security_value	Monetary SFT legs.
security_haircut, fx_haircut	Haircuts.
rwea_pool_irb_ul, el_pool_irb, rwea_pool_sa, pool_ead	Pool inputs.
ka, pool_k	Pool capital ratio.
attachment, detachment	Tranche attachment and detachment points.
p	Supervisory SSFA parameter.
floor	Minimum risk weight.
pool_type	Pool category.
senior, sts, resecuritisation	Logical tranche flags.
effective_number	Effective number of pool exposures.
average_lgd	Average pool LGD.
tranche_maturity, maturity	Maturity in years.
cqs	Credit-quality step.
approach	Securitisation approach.
items	BA-CVA rows, each with the documented nine numeric fields.
days_late	Settlement delay in days.

Value

Numeric formula result or named vector.

See Also

analyze_counterparty_risk(), analyze_securitisation()

Examples

```
sa_ccr_exposure_value(100, 30, 20, 1.4)
sft_exposure_value(100, 90, 0.1, 0.05)
```

`sa_exposure_value` *Standardised Credit-risk and CRM Formulae*

Description

Granular, side-effect-free formulae for exposure value, risk weights, real estate treatment and credit-risk mitigation. Rates and weights are decimals. Monetary inputs and outputs use one caller-selected currency consistently.

Usage

```
sa_exposure_value(
  gross_carrying_amount,
  specific_adjustments = 0,
  additional_valuation_adjustments = 0,
  other_own_funds_reductions = 0,
  committed_undrawn = 0,
  annex_i_class,
  data_path = "GROSS_COMPONENTS",
  net_carrying_amount_article_111 = NULL,
  parameters = NULL
)

sa_risk_weight(
  exposure_class,
  cqs = NULL,
  short_term = FALSE,
  transactor = FALSE,
  retail_eligible = FALSE,
  defaulted = FALSE,
  default_coverage_ratio = 0,
  specialised_lending_type = "",
  parameters = NULL
)

real_estate_risk_weight(
  ead,
  property_value,
  property_type,
  ipre,
  counterparty_rw,
  senior_liens = 0,
  adc = FALSE,
```

```
  parameters = NULL
)

crm_maturity_factor(t_protection, t_exposure, minimum_t = 0.25, maximum_T = 5)

crm_adjusted_exposure(exposure, collateral, he, hc, hfx)
```

Arguments

gross_carrying_amount, specific_adjustments, additional_valuation_adjustments, other_own_funds_reductions	Monetary amounts.
committed_undrawn	Undrawn commitment amount.
annex_i_class	CRR Annex I conversion-factor class.
data_path	"GROSS_COMPONENTS" or "NET_ARTICLE_111".
net_carrying_amount_article_111	Optional net carrying amount.
parameters	Optional parameter data frame or rwa_parameter_store; bundled regulatory parameters are used by default.
exposure_class, cqs	Exposure class and credit-quality step.
short_term, transactor, retail_eligible, defaulted	Logical classification flags.
default_coverage_ratio	Coverage ratio for defaulted exposures.
specialised_lending_type	Optional specialised-lending category.
ead, property_value, senior_liens	Monetary real-estate inputs.
property_type	Property category.
ipre, adc	Logical real-estate flags.
counterparty_rw	Counterparty risk weight.
t_protection, t_exposure, minimum_t, maximum_T	Maturity-mismatch inputs in years.
exposure, collateral	Exposure and collateral values.
he, hc, hfx	Exposure, collateral and currency haircuts.

Value

Numeric value, named numeric vector, or documented list depending on the formula.

See Also

regulatory_parameters(), override_regulatory_parameters()

Examples

```
sa_exposure_value(100, 2, 0, 0, 40, "CLASS_2", "GROSS_COMPONENTS")
crm_adjusted_exposure(100, 60, 0.05, 0.10, 0.08)
```

---

select_rule_set *Select a Rule Set*

Description

Returns a modified copy of canonical tables with the requested rule set in run_config; it never mutates the caller's object.

Usage

```
select_rule_set(tables, rule_set_id)
```

Arguments

tables	Canonical input tables.
rule_set_id	Identifier listed by available_rule_sets().

Value

Modified canonical tables.

Examples

```
t <- generate_synthetic_tables(bank_profile = "KSA_BANK")
t <- select_rule_set(t, "CRR3-EU-FL")
```

---

table_dictionary *Canonical Table Dictionary*

Description

Canonical Table Dictionary

Usage

```
table_dictionary()
```

Value

A data frame describing all logical tables, workbooks and sheets.

See Also

`table_schema()`

Examples

```
head(table_dictionary())
```

`table_schema` *Canonical Table Schema*

Description

Canonical Table Schema

Usage

```
table_schema(table)
```

Arguments

`table`	Logical table name from `table_dictionary()`.

Value

A list with workbook, sheet, Excel table, required flag and columns.

Examples

```
table_schema("exposure_lot")
```

validate_dataset *Validate a Canonical Dataset*

Description

Reads and validates all canonical input workbooks without calculating or writing any output files.

Usage

```
validate_dataset(dataset)
```

Arguments

dataset	Dataset directory containing an inputs subdirectory.

Value

An rwa_validation_report object.

See Also

calculate_dataset(), generate_synthetic_dataset()

Examples

```
root <- file.path(tempdir(), "rwa-validation-example")
dataset <- generate_synthetic_dataset(root, bank_profile = "KSA_BANK")
as.data.frame(validate_dataset(dataset))
```

Index