Package 'financialinstrumentsaccounting'

November 28, 2026

Title Financial Instruments Accounting Reference Engine

Version 0.1.1

Description Native, granular calculations for financial-instrument accounting, including classification, measurement, effective interest, impairment, scenario-weighted expected credit losses, modifications, derecognition, hedge accounting, journals, disclosures, model backtesting, reconciliation, controls, and audit lineage. Provides explicit policy inputs and reproducible synthetic reference profiles. It is an independent reference implementation and does not constitute audited accounting software or authoritative accounting advice.

License Apache License (== 2.0)

Encoding UTF-8

Depends R (>= 4.1.0)

Imports digest, jsonlite

Suggests knitr, pkgbuild, pkgload, rcmdcheck, rmarkdown, roxygen2, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/rds0001/financial-instruments-accounting-r

BugReports https://github.com/rds0001/financial-instruments-accounting-r/issues

Config/roxygen2/version 8.1.1

NeedsCompilation no

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

Maintainer Dimitrios Geromichalos <riskdatascience@web.de>

Repository CRAN

Date/Publication 2026-11-28 15:20:26 UTC

Contents

financialinstrumentsaccounting-package 2
classification_decisions 3
core_formulas 6
reference_data 9
reporting_monitoring 11
risk_event_hedge 14
workflow 17

Index **19**

`financialinstrumentsaccounting-package`

Financial Instruments Accounting

Description

`financialinstrumentsaccounting` provides 80 granular, native R functions for financial-instrument classification, measurement, effective interest, impairment, hedge accounting, journals, disclosures, model monitoring, reconciliation, controls, and audit lineage. It also provides a controlled end-to-end reference workflow over two fully synthetic offline profiles.

Details

Formula functions accept ordinary R vectors, matrices, and lists. Domain functions accept named canonical rows and tables. The workflow uses the same public functions, returns an `fae_analysis_result`, and exposes 30 named result tables, 28 controls, parameters, metrics, and lineage through public accessors. Rates and probabilities use decimal notation; time is in years; monetary inputs in one calculation use one currency unless an explicit FX mapping is supplied. Missing judgements, approvals, and model inputs are rejected rather than silently inferred.

Where to start

- Use `discount_factor()`, `effective_interest_rate()`, and `component_ecl()` for independent formula calculations.
- Use `assess_sppi()`, `classify_instrument()`, and `assess_stage()` for row-level accounting decisions.
- Use `load_reference_data()` and `run_dataset()` for a complete synthetic reference run, then inspect it with `get_metrics()`, `get_table()`, `get_controls()`, and `get_lineage()`.
- Read `vignette("getting-started")` first; the other vignettes cover impairment, hedges, journals, disclosures, reference data, and controls.

Reference data and status

`SMALL_SA_RETAIL_FIA` and `MID_SIZE_UNIVERSAL_FIA` are deterministic, synthetic regression fixtures. `APPROVED_REFERENCE` means that the package's technical controls passed. It is not an institutional, audit, or production approval.

Legal and professional scope

This independent package is not affiliated with or endorsed by the IFRS Foundation or the International Accounting Standards Board. It is reference software under Apache License 2.0. It does not provide accounting, audit, legal, investment, prudential, or tax advice and does not replace authoritative standards, entity policies, professional judgement, model validation, or financial-close controls.

Author(s)

Maintainer: Dimitrios Geromichalos <riskdatascience@web.de>

Authors:

- RiskDataScience GmbH [copyright holder]

References

https://www.ifrs.org/issued-standards/list-of-standards/ifrs-9-financial-instruments/

See Also

core_formulas, classification_decisions, risk_event_hedge, reporting_monitoring, reference_data, workflow

Examples

```
discount_factor(0.05, 1)
result <- run_dataset(load_reference_data("SMALL_SA_RETAIL_FIA"))
stopifnot(
  result$status == "APPROVED_REFERENCE",
  length(get_table(result, "DISCLOSURE_FACTS")) > 0L,
  all(vapply(get_controls(result), function(x) x$status == "PASS", logical(1)))
)
```

---

classification_decisions

Classification, Scope, Measurement and Staging Decisions

Description

Native row-level decisions for scope and recognition, business-model and contractual cash-flow assessment, classification, initial/fair-value measurement, own-credit presentation, interest revenue, impairment scope, staging and controlled IBOR interfaces.

Usage

```
assess_scope(instrument)

assess_recognition(instrument)

assess_business_model(model)

assess_sppi(features)

classify_instrument(instrument, features, scope_decision = "IN_IFRS9_SCOPE")

assess_reclassification(instrument, features)

initial_measurement(instrument, category)

split_own_credit(market, category)

present_fair_value(instrument, market, category)

interest_revenue(instrument, stage, allowance)

impairment_scope(instrument)

assess_stage(facts, policy = NULL)

assess_sicr(facts, policy = NULL)

stage_movements(stage_results)

assess_ibor_relief(applicable, policy_reference, approval_reference)
```

Arguments

instrument	Named list of canonical instrument facts. Monetary amounts use the instrument's currency; required fields depend on the function.
model	Approved business-model judgement: HOLD_TO_COLLECT, HOLD_TO_COLLECT_AND_SELL, or OTHER.
features	Named list containing the eight documented basic-lending checks, leverage flag and any explicitly approved override.
scope_decision	Explicit scope result, normally IN_IFRS9_SCOPE.
category	Classification category used for measurement presentation.
market	Named list of supplied fair-value and valuation-control facts.
stage	Explicit impairment stage.
allowance	Allowance in the instrument's currency; signed POCI changes are permitted.
facts	Named staging facts including instrument ID, impairment scope, days past due and origination/current ratings.

policy	Explicit versioned policy list. NULL selects the fixed packaged policy EU_FINANCIAL_INSTRUMENTS_2026_
stage_results	List of rows with prior_stage and stage.
applicable	One non-missing logical IBOR applicability judgement.
policy_reference, approval_reference	Non-empty identifiers for the institution's policy and case approval.

Details

These functions validate supplied facts and approved judgements; they do not infer a business model, designation, override or institutional approval. Initial measurement adds transaction costs for financial assets outside FVTPL and subtracts them for liabilities. Interest is calculated on gross carrying amount in stages 1/2, net amount in stage 3 and the credit-adjusted net basis for POCI. The 30- and 90-day thresholds remain rebuttable/policy-driven and credit impairment takes precedence over SICR.

Value

A named list containing the requested decision, amount, reason, formula/rule ID and source reference. stage_movements() returns a list of rows with prior stage, current stage and instrument count.

References

International Accounting Standards Board (2014), IFRS 9, paragraphs 2.1–2.7, 3.1, 4.1–4.2, 5.1, 5.4, 5.5 and 5.7; amendments on classification and measurement applicable in the packaged 2026 policy. Official source index: https://www.ifrs.org/issued-standards/list-of-standards/ifrs-9-financial-instruments/.

See Also

core_formulas, scenario_ecl(), reference_data

Examples

```
instrument <- list(
  instrument_id="x", instrument_type="LOAN", scope_assessment="IN_IFRS9_SCOPE",
  recognition_status="RECOGNISED", trade_date_policy="TRADE_DATE",
  regular_way=FALSE, business_model="HOLD_TO_COLLECT", liability=FALSE,
  equity=FALSE, fair_value=100, transaction_costs=2, transaction_price=100,
  impairment_scope="GENERAL_ECL", gross_amount=100, eir=0.1
)
features <- list(
  principal_consistent=TRUE, basic_interest=TRUE, leverage=FALSE,
  modified_time_value_pass=TRUE, prepayment_compensation_reasonable=TRUE,
  extension_basic=TRUE, non_recourse_pass=TRUE, cli_lookthrough_pass=TRUE,
  contingent_feature_basic=TRUE, business_model_change=FALSE
)
assess_scope(instrument)
assess_recognition(instrument)
assess_business_model("HOLD_TO_COLLECT")
```

```
assess_sppi(features)
classify_instrument(instrument, features)
assess_reclassification(instrument, features)
initial_measurement(instrument, "AMORTISED_COST")
market <- list(instrument_id="x", fair_value=110, total_fv_change=10,
               own_credit_change=4, ifrs13_level="2", valuation_model="DCF",
               valuation_status="APPROVED")
present_fair_value(instrument, market, "FVOCI_EQUITY")
split_own_credit(market, "FVTPL_LIABILITY")
interest_revenue(instrument, "STAGE_3", 20)
impairment_scope(instrument)
stage_facts <- list(instrument_id="x", impairment_scope="GENERAL_ECL",
                    days_past_due=0, rating_orig=3, rating_current=3)
assess_stage(stage_facts)
assess_sicr(stage_facts)
stage_movements(list(list(prior_stage="STAGE_1", stage="STAGE_2")))
assess_ibor_relief(FALSE, "IBOR_PHASE2_POLICY_V1", "CASE_APPROVAL_1")
```

`core_formulas` *Core Financial-Instrument Accounting Formulas*

Description

Pure deterministic calculations for discounting, effective interest, amortised cost, probability curves, credit loss, recoveries, modifications, hedges and coverage. Rates and probabilities are decimals; time is measured in years; monetary values within one call use one currency.

Usage

```
discount_factor(rate, year_fraction)

effective_interest_rate(initial_net, cashflows, tolerance = 1e-12)

amortised_cost_step(opening, eir, time, cash_received, adjustment = 0)

amortisation_schedule(opening, eir, cashflows)

marginal_pd_from_cumulative(cumulative)

survival_from_pd(marginal)

cumulative_pd_from_conditional(conditional)

component_ecl(marginal_pd, ead, lgd, eir, year)

recovery_present_value(recoveries, eir)

cash_shortfall_ecl(contractual, expected, eir)
```

```
modification_gain_loss(gross_before, modified_cashflows, original_eir)

hedge_ineffectiveness(hedging_change, hedged_change)

ead_profile(drawn, undrawn, ccf, prepayment)

lgd_from_recoveries(ead, recoveries, eir, costs = list())

provision_matrix_ecl(exposure, loss_rate)

weighted_ecl(scenario_losses, probabilities)

poci_allowance_change(current_lifetime_ecl, initial_lifetime_ecl)

liability_modification_test(
  original_present_value,
  modified_present_value,
  qualitative_substantial = FALSE,
  tolerance = 0.1
)

coverage_ratio(allowance, gross_exposure)

benchmark_ecl(iterations = 10000L)
```

Arguments

`rate`, `eir`, `original_eir`	Annual effective rate as a decimal, strictly greater than `-1` where discounting is performed.
`year_fraction`, `time`, `year`	Non-negative time in years.
`initial_net`	Positive initial net carrying amount.
`cashflows`, `contractual`, `expected`, `modified_cashflows`, `recoveries`, `costs`	Two-column matrix/data frame or list of `(time, amount)` pairs. Times and amounts are finite and non-negative.
`tolerance`	Positive numerical tolerance. For `liability_modification_test()`, the policy threshold in `[0, 1]`.
`opening`, `gross_before`, `ead`, `exposure`, `allowance`, `gross_exposure`	Monetary amounts in the call's currency. Required non-negative unless the function expressly returns a signed result.
`cash_received`	Non-negative receipt in the period.
`adjustment`	Signed period adjustment.
`cumulative`	Non-decreasing cumulative default probabilities in `[0,1]`.
`marginal`	Unconditional marginal default probabilities in `[0,1]`.

`conditional` Conditional period default probabilities in `[0,1]`.

`marginal_pd`, `lgd`, `loss_rate`
: Probabilities or rates in `[0,1]`.

`hedging_change`, `hedged_change`
: Signed fair-value changes; gains are positive.

`drawn`, `undrawn` Non-negative exposure vectors.

`ccf`, `prepayment` Equal-length rate vectors in `[0,1]`.

`scenario_losses`
: Named non-negative scenario losses.

`probabilities` Named scenario probabilities whose keys match losses and whose sum is one within `1e-9`.

`current_lifetime_ecl`, `initial_lifetime_ecl`
: Non-negative lifetime loss estimates; the returned POCI movement remains signed.

`original_present_value`, `modified_present_value`
: Non-negative present values measured using the original effective interest rate.

`qualitative_substantial`
: One non-missing logical management judgement.

`iterations` Integer count from 1 through 1,000,000.

Details

No function rounds monetary results. Inputs must be finite and missing values are rejected. `effective_interest_rate()` uses a bracketed bisection root over `[-0.999999, 10]`, with at most 250 iterations. `component_ecl()` discounts PD times EAD times LGD to the supplied default time. Scenario losses must be calculated separately before `weighted_ecl()` is called. Policy judgements, model calibration and data approval remain caller responsibilities.

Value

Scalar formula functions return one numeric value. Probability and EAD curve functions return numeric vectors in input order. `amortisation_schedule()` returns a data frame with `time`, `opening`, `interest`, `cash_received`, and `closing`. The liability modification test returns a named list with `ratio`, `substantial`, and `threshold`. `coverage_ratio()` returns `NA_real_` for a zero denominator. The benchmark returns iterations, checksum, elapsed seconds, peak bytes (unavailable and therefore `NA_real_` in portable R), and formula identifier.

References

International Accounting Standards Board (2014), IFRS 9, paragraphs 5.4.1, 5.5.17, B3.3.6 and Appendix A (effective interest and expected credit losses). See the independently maintained official source index at https://www.ifrs.org/issued-standards/list-of-standards/ifrs-9-financial-instruments/.

See Also

`scenario_ecl()`, `interest_revenue()`, `process_modification()`

Examples

```
discount_factor(0.05, 1)
stopifnot(abs(effective_interest_rate(1000, list(c(1, 1100))) - 0.1) < 1e-10)
amortised_cost_step(1000, 0.1, 1, 200)
amortisation_schedule(1000, 0.1, list(c(1, 200), c(2, 990)))
marginal_pd_from_cumulative(c(0.1, 0.3))
survival_from_pd(c(0.1, 0.2))
cumulative_pd_from_conditional(c(0.1, 0.2))
component_ecl(0.02, 100000, 0.4, 0.05, 1)
cash_shortfall_ecl(list(c(1, 105)), list(c(2, 55.125)), 0.05)
modification_gain_loss(100, list(c(1, 99)), 0.1)
hedge_ineffectiveness(100, -90)
ead_profile(100, 50, 0.4, 0.1)
recovery_present_value(list(c(2, 121)), 0.1)
lgd_from_recoveries(100, list(c(1, 55)), 0.1)
provision_matrix_ecl(1000, 0.03)
weighted_ecl(c(base = 10, stress = 30), c(base = 0.75, stress = 0.25))
poci_allowance_change(20, 30)
liability_modification_test(100, 110)
coverage_ratio(20, 100)
benchmark_ecl(10)
```

---

reference_data *Packaged Policies, Schemas and Synthetic Reference Profiles*

Description

Access versioned policy parameters, formula metadata, source metadata, canonical table schemas and two complete synthetic input profiles. Validate a dataset or select the unique official bitemporal record at explicit reporting and knowledge cut-offs. All returned objects are independent R lists and no function performs a network request.

Usage

```
get_policy(policy_id = "EU_FINANCIAL_INSTRUMENTS_2026_V1")

get_parameters(policy_id = "EU_FINANCIAL_INSTRUMENTS_2026_V1")

get_schema(table = NULL)

get_formula(formula_id, rule_set_id = "FIA_RULES_2026_V1")

get_sources()

list_profiles()

load_reference_data(profile_id)
```

```
export_profile(profile_id, destination)

select_snapshot(tables, reporting_date, knowledge_time)

validate_dataset(tables, reporting_date = NULL, knowledge_time = NULL)
```

Arguments

policy_id	Explicit accounting-policy identifier. Only the packaged EU_FINANCIAL_INSTRUMENTS_2026_V1 policy is accepted.
table	Optional canonical table name. NULL returns all 19 schemas.
formula_id	Formula registry identifier such as "F-ECL-001".
rule_set_id	Explicit rule-set identifier. Only FIA_RULES_2026_V1 is packaged.
profile_id	One value returned by list_profiles().
destination	A new directory path. Existing paths are rejected.
tables	Named list containing canonical row lists for all 19 tables, or a subset for select_snapshot().
reporting_date	ISO date (YYYY-MM-DD) used for the valid-time cut-off.
knowledge_time	ISO timestamp with explicit UTC offset used for the transaction-time cut-off.

Details

Resources are synthetic and bundled for deterministic offline use. select_snapshot() requires exactly one ACTIVE, official version per business key whose valid and known intervals include both cut-offs. validate_dataset() checks the exact table/column contract, primary keys, typed numeric/logical fields, supported view, scenario weights, curve keys and positive FX rates before returning the selected snapshot. It raises a condition inheriting from fae_validation_error on rejection.

Value

get_policy() and get_parameters() return a named parameter list; get_schema() returns one schema or all schemas; get_formula() returns one registry row; get_sources() returns a list of source-metadata rows; list_profiles() returns two stable character identifiers; load_reference_data(), select_snapshot() and validate_dataset() return named lists of row lists; export_profile() invisibly returns the normalized destination path after writing 19 JSON table files and a JSON manifest.

References

International Accounting Standards Board (2014), IFRS 9, paragraphs 5.5.3–5.5.17. Official source index: https://www.ifrs.org/issued-standards/list-of-standards/ifrs-9-financial-instruments/.

See Also

run_dataset(), core_formulas

Examples

```
list_profiles()
small <- load_reference_data("SMALL_SA_RETAIL_FIA")
length(small$instruments)
get_schema("instruments")$columns[1:3]
get_policy()$past_due_sicr_days
get_parameters()$default_backstop_days
get_formula("F-ECL-001")$name
length(get_sources())
selected <- select_snapshot(
  list(instruments = small$instruments), "2026-12-31",
  "2026-09-02T12:00:00+02:00"
)
stopifnot(length(selected$instruments) == 5L)
validated <- validate_dataset(small)
stopifnot(length(validated) == 19L)
target <- tempfile("fae-profile-")
export_profile("SMALL_SA_RETAIL_FIA", target)
unlink(target, recursive = TRUE)
```

---

reporting_monitoring *Journals, Rollforwards, Disclosures and Model Monitoring*

Description

Build balanced reference journals, reconcile balances, produce allowance and gross-carrying-amount rollforwards, create atomic and aggregated disclosure facts, monitor model observations, compare challengers, analyse scenario sensitivity and keep regulatory expected loss separate from accounting ECL.

Usage

```
build_journals(
  ecl_results,
  opening_allowances,
  classifications,
  instruments,
  processed_events = list(),
  own_credit_results = list(),
  hedge_results = list()
)

allowance_rollforward(instruments, opening, ecl_results, processed_events, fx)

gross_rollforward(instruments, opening, processed_events, fx)

reconcile_ledger(expected, actual, tolerance = 1e-09)
```

```
disclosure_facts(
  instruments,
  ecl_results,
  classifications,
  stages,
  fx,
  presentation_currency
)

aggregate_disclosures(facts)

backtest_pd(observations)

backtest_lgd(observations)

backtest_ead(observations)

backtest_staging(observations)

backtest_models(observations)

scenario_sensitivity(losses, probabilities)

compare_challenger(reference, challenger)

regulatory_bridge(instruments, ecl_results, stages, fx)
```

Arguments

`ecl_results`	List of instrument ECL summary rows.
`opening_allowances`, `opening`	Named numeric opening balances by instrument.
`classifications`, `stages`	Named character mappings by instrument.
`instruments`	List of canonical instrument rows.
`processed_events`, `own_credit_results`, `hedge_results`	Lists of processed event, own-credit or hedge rows.
`fx`	Named positive presentation-currency FX rates by instrument.
`expected`, `actual`	Scalars in the same unit.
`tolerance`	Non-negative absolute reconciliation tolerance.
`presentation_currency`	Explicit presentation-currency code.
`facts`	Atomic disclosure rows produced by `disclosure_facts()`.
`observations`	Approved model-monitoring rows with positive exposure weights and explicit owner, validation status and approval ID.

`losses` Named non-negative standalone scenario losses.

`probabilities` Named scenario probabilities summing to one.

`reference, challenger`
Monitoring result rows keyed by model type and segment.

Details

Journals are balanced reference entries and never post to a ledger. Institutional account mapping and approval remain external. Disclosure coverage is aggregate allowance divided by aggregate gross amount. Regulatory inputs are only displayed in a separate bridge and never feed the accounting estimate. Model diagnostics and challenger comparisons are management analyses without accounting effects.

Value

Table functions return lists of named rows. `reconcile_ledger()` returns expected, actual, signed difference, tolerance and PASS/FAIL. Journals contain batch, instrument, account, debit, credit and movement. Backtests contain weighted prediction/actual, calibration ratio, mean absolute error, bias and, where identifiable, pairwise AUC.

References

International Accounting Standards Board (2014), IFRS 9, paragraphs 5.5 and 6.5; IFRS 7 paragraphs 35A–35N. Official source index: https://www.ifrs.org/issued-standards/list-of-standards/ifrs-7-financial-instruments-disclosures/.

See Also

scenario_ecl(), risk_event_hedge, run_dataset()

Examples

```
instruments <- list(list(instrument_id="x", gross_amount=100, entity_id="E",
                  portfolio="P", currency="EUR", regulatory_expected_loss=15))
ecl <- list(list(instrument_id="x", stage="STAGE_1", final_ecl=20))
build_journals(ecl, c(x=5), c(x="AMORTISED_COST"), instruments)
allowance_rollforward(instruments, c(x=5), ecl, list(), c(x=1))
gross_rollforward(instruments, c(x=90), list(), c(x=1))
reconcile_ledger(100, 99)
facts <- disclosure_facts(instruments, ecl, c(x="AMORTISED_COST"),
                          c(x="STAGE_1"), c(x=1), "EUR")
aggregate_disclosures(facts)
obs <- list(list(model_type="PD", model_version_id="v1", predicted_value=.2,
            actual_value=.1, exposure_weight=1, outcome_flag=FALSE,
            model_owner="Risk", validation_status="VALIDATED", approval_id="A",
            segment="X"))
backtest_pd(obs)
backtest_lgd(modifyList(obs[[1]], list(model_type="LGD")) |> list())
backtest_ead(modifyList(obs[[1]], list(model_type="EAD")) |> list())
backtest_staging(modifyList(obs[[1]], list(model_type="STAGING")) |> list())
backtest_models(obs)
```

```
scenario_sensitivity(c(base=10,stress=30),c(base=.75,stress=.25))
compare_challenger(list(list(model_type="PD",segment="X",mean_absolute_error=.1,bias=.02)),
               list(list(model_type="PD",segment="X",mean_absolute_error=.05,bias=.01)))
regulatory_bridge(instruments,ecl,c(x="STAGE_1"),c(x=1))
```

`risk_event_hedge` *Impairment, Overlay, Event and Hedge Analyses*

Description

Calculate scenario-based credit losses, govern credit enhancements and overlays, process derecognition/modification/write-off/recovery events, and measure documented hedge relationships and OCI reserves.

Usage

```
allocate_credit_enhancements(collateral, instruments)

scenario_ecl(
  instrument,
  stage,
  pd_curves,
  lgd_profiles,
  ead_profiles,
  probabilities,
  cashflows = list(),
  reporting_date = NULL
)

validate_overlay(overlay, reporting_date)

allocate_overlay(amount, allocation_weights)

reconcile_overlay(amount, allocations, tolerance = 1e-09)

assess_derecognition(event, instrument)

continuing_involvement(event, instrument)

process_modification(event, instrument)

process_writeoff(event, instrument)

process_recovery(event, instrument)

measure_hedge(relationship)

validate_hedge_designation(relationship)
```

```
assess_hedge_effectiveness(relationship)

oci_rollforward(
  opening,
  fair_value_movement = 0,
  effective_hedge_movement = 0,
  costs = 0,
  recycling = 0,
  basis_adjustment = 0
)

hedge_reserve_rollforward(opening, hedge_results)
```

Arguments

collateral, instruments	List of canonical collateral or instrument rows.
instrument	Named canonical instrument row.
stage	Explicit impairment stage.
pd_curves, lgd_profiles, ead_profiles	Lists of rows keyed by scenario_id and positive integer year, with marginal PD, LGD and EAD.
probabilities	Named scenario probabilities summing to one.
cashflows	Optional canonical contractual and expected cash-flow rows.
reporting_date	ISO reporting date required when an overlay is nonzero.
overlay	Named governed overlay row.
amount	Signed approved overlay amount or non-negative event amount, depending on the function.
allocation_weights	Named allocation weights summing to one.
allocations	Named signed allocated amounts.
tolerance	Non-negative absolute reconciliation tolerance.
event	Named approved event facts.
relationship	Named hedge relationship and lifecycle facts; changes are cumulative and monetary amounts use one currency.
opening	Signed opening OCI/hedge reserve.
fair_value_movement, effective_hedge_movement, costs, recycling, basis_adjustment	Signed reserve movements in one currency.
hedge_results	List of measured hedge rows.

Details

Stage 1 limits default events to year one; lifetime stages include every supplied annual curve point. Scenario losses are calculated before probability weighting. Direct cash-shortfall and provision-matrix methods remain explicit. Overlays require owner, thesis, approval, expiry and a double-count check. Event and hedge decisions consume documented inputs and never infer approvals or post to a ledger. Hedge effectiveness uses the lower-of absolute changes for cash-flow and net-investment hedges.

Value

`scenario_ecl()` returns `list(details, summary)`, where `details` contains component rows and `summary` contains weighted, overlay and final ECL. Allocation and event functions return row lists; hedge functions return one result row. Rollforwards return a named reserve row.

References

International Accounting Standards Board (2014), IFRS 9, paragraphs 3.2–3.3, 5.5.17, B5.5.28–B5.5.55 and 6.4–6.6. Official source index: https://www.ifrs.org/issued-standards/list-of-standards/ifrs-9-financial-instruments/.

See Also

core_formulas, classification_decisions, `build_journals()`

Examples

```
collateral <- list(list(collateral_id="C", instrument_id="x", fair_value=100,
                        haircut=0.2, cost=5, allocated_amount=70, recovery_year=2))
allocate_credit_enhancements(collateral, list(list(instrument_id="x")))
instrument <- list(instrument_id="x", impairment_scope="GENERAL_ECL",
                   ecl_method="COMPONENT_PD_LGD_EAD", eir=0.05, overlay=0)
pd <- list(list(scenario_id="base", year=1, marginal_pd=0.02))
lgd <- list(list(scenario_id="base", year=1, lgd=0.4))
ead <- list(list(scenario_id="base", year=1, ead=100000))
scenario_ecl(instrument, "STAGE_1", pd, lgd, ead, c(base=1))$summary$final_ecl
governed <- list(overlay_id="O", overlay_owner="Risk", overlay_approved=TRUE,
                 overlay_expiry="2026-12-31", overlay_thesis="Emerging risk",
                 overlay_double_count_checked=TRUE, overlay=10)
validate_overlay(governed, "2026-12-31")
allocate_overlay(10, c(x=0.3, y=0.7))
reconcile_overlay(10, c(x=3, y=7))
source <- list(instrument_id="x", gross_amount=100)
transfer <- list(event_id="E", instrument_id="x", event_type="ASSET_TRANSFER",
                 approved=TRUE, reason_code="APPROVED", amount=10,
                 risks_rewards_transferred=TRUE)
assess_derecognition(transfer, source)
continuing_involvement(modifyList(transfer, list(risks_rewards_transferred=FALSE,
                      control_retained=TRUE, continuing_involvement_amount=4)), source)
process_modification(list(event_id="M", instrument_id="x",
                     event_type="ASSET_MODIFICATION_NO_DERECOGNITION", approved=TRUE,
                     reason_code="APPROVED", amount=0, modified_present_value=90,
```

```
                    derecognition_approved=FALSE), source)
process_writeoff(list(event_id="W", instrument_id="x", event_type="WRITEOFF",
                approved=TRUE, reason_code="APPROVED", amount=10), source)
process_recovery(list(event_id="R", instrument_id="x", event_type="RECOVERY",
                approved=TRUE, reason_code="APPROVED", amount=10), source)
hedge <- list(relationship_id="H", hedge_type="CASH_FLOW", hedging_change=120,
              hedged_change=-100, hedge_ratio=1, target_hedge_ratio=1,
              hedging_instrument_eligible=TRUE, hedged_item_eligible=TRUE,
              risk_component_eligible=TRUE, designation_documented=TRUE,
              risk_management_objective="Manage risk", designated=TRUE,
              economic_relationship=TRUE, credit_risk_dominates=FALSE,
              forecast_transaction_expected=TRUE)
validate_hedge_designation(hedge)
assess_hedge_effectiveness(hedge)
measured <- measure_hedge(hedge)
hedge_reserve_rollforward(10, list(measured))
oci_rollforward(10, fair_value_movement=5, recycling=2, basis_adjustment=1)
```

`workflow` *Complete In-Memory Accounting Workflow and Result Accessors*

Description

Validate and calculate a complete canonical dataset using the same native R functions exposed for granular analysis. Access named output tables, metrics, controls and audit lineage, or export a read-only JSON result package.

Usage

```
run_dataset(dataset, policy = NULL)

get_metrics(result)

get_table(result, name)

get_controls(result)

get_lineage(result)

export_results(result, destination)
```

Arguments

`dataset`	A named canonical 19-table list or a directory created by `export_profile()` containing one JSON file per table.
`policy`	Explicit versioned policy list. `NULL` selects the packaged `EU_FINANCIAL_INSTRUMENTS_2026_V1` policy.
`result`	An object returned by `run_dataset()`.

name	Exact book/sheet result key or a unique sheet name.
destination	A new directory path. Existing paths are rejected.

Details

A rejected input never yields a partially approved result. Run control and policy identifiers/dates must agree. The workflow performs no network access, journal posting or mutation of inputs. APPROVED_REFERENCE means all embedded controls passed; it is not an institutional or official accounting approval. Lineage includes deterministic input, policy, rule, code and resource hashes and contract version 1.1.0.

Value

run_dataset() returns an immutable-by-contract S3 object of class fae_analysis_result containing tables, metrics, lineage, status and official=FALSE. Accessors return deep copies. export_results() invisibly returns the normalized destination after writing seven grouped JSON files, results.json and a SHA-256 manifest.

References

International Accounting Standards Board (2014), IFRS 9 and IFRS 7. Official source indexes: https://www.ifrs.org/issued-standards/.

See Also

reference_data, classification_decisions, risk_event_hedge, reporting_monitoring

Examples

```
result <- run_dataset(load_reference_data("SMALL_SA_RETAIL_FIA"))
result$status
get_metrics(result)$opening_gross_exposure
length(get_table(result,"STAGING"))
all(vapply(get_controls(result), function(x) x$status == "PASS", logical(1)))
get_lineage(result)$contract_version
target <- tempfile("fae-results-")
export_results(result,target)
unlink(target,recursive=TRUE)
```

Index

∗ **package**
financialinstrumentsaccounting-package, 2

aggregate_disclosures *(*reporting_monitoring*)*, 11
allocate_credit_enhancements *(*risk_event_hedge*)*, 14
allocate_overlay *(*risk_event_hedge*)*, 14
allowance_rollforward *(*reporting_monitoring*)*, 11
amortisation_schedule *(*core_formulas*)*, 6
amortised_cost_step *(*core_formulas*)*, 6
assess_business_model *(*classification_decisions*)*, 3
assess_derecognition *(*risk_event_hedge*)*, 14
assess_hedge_effectiveness *(*risk_event_hedge*)*, 14
assess_ibor_relief *(*classification_decisions*)*, 3
assess_reclassification *(*classification_decisions*)*, 3
assess_recognition *(*classification_decisions*)*, 3
assess_scope *(*classification_decisions*)*, 3
assess_sicr *(*classification_decisions*)*, 3
assess_sppi *(*classification_decisions*)*, 3
assess_sppi(), *2*
assess_stage *(*classification_decisions*)*, 3
assess_stage(), *2*

backtest_ead *(*reporting_monitoring*)*, 11
backtest_lgd *(*reporting_monitoring*)*, 11
backtest_models *(*reporting_monitoring*)*, 11
backtest_pd *(*reporting_monitoring*)*, 11
backtest_staging *(*reporting_monitoring*)*, 11
benchmark_ecl *(*core_formulas*)*, 6
build_journals *(*reporting_monitoring*)*, 11
build_journals(), *16*

cash_shortfall_ecl *(*core_formulas*)*, 6
classification_decisions, *3*, 3, *16*, *18*
classify_instrument *(*classification_decisions*)*, 3
classify_instrument(), *2*
compare_challenger *(*reporting_monitoring*)*, 11
component_ecl *(*core_formulas*)*, 6
component_ecl(), *2*
continuing_involvement *(*risk_event_hedge*)*, 14
core_formulas, *3*, *5*, 6, *10*, *16*
coverage_ratio *(*core_formulas*)*, 6
cumulative_pd_from_conditional *(*core_formulas*)*, 6

disclosure_facts *(*reporting_monitoring*)*, 11
discount_factor *(*core_formulas*)*, 6
discount_factor(), *2*

ead_profile *(*core_formulas*)*, 6
effective_interest_rate *(*core_formulas*)*, 6
effective_interest_rate(), *2*
export_profile *(*reference_data*)*, 9
export_results *(*workflow*)*, 17

financialinstrumentsaccounting *(*financialinstrumentsaccounting-package*)*, 2
financialinstrumentsaccounting-package, 2

get_controls (workflow), 17
get_controls(), 2
get_formula (reference_data), 9
get_lineage (workflow), 17
get_lineage(), 2
get_metrics (workflow), 17
get_metrics(), 2
get_parameters (reference_data), 9
get_policy (reference_data), 9
get_schema (reference_data), 9
get_sources (reference_data), 9
get_table (workflow), 17
get_table(), 2
gross_rollforward
(reporting_monitoring), 11

hedge_ineffectiveness (core_formulas), 6
hedge_reserve_rollforward
(risk_event_hedge), 14

impairment_scope
(classification_decisions), 3
initial_measurement
(classification_decisions), 3
interest_revenue
(classification_decisions), 3
interest_revenue(), 8

lgd_from_recoveries (core_formulas), 6
liability_modification_test
(core_formulas), 6
list_profiles (reference_data), 9
load_reference_data (reference_data), 9
load_reference_data(), 2

marginal_pd_from_cumulative
(core_formulas), 6
measure_hedge (risk_event_hedge), 14
modification_gain_loss (core_formulas),
6

oci_rollforward (risk_event_hedge), 14

poci_allowance_change (core_formulas), 6
present_fair_value
(classification_decisions), 3
process_modification
(risk_event_hedge), 14
process_modification(), 8
process_recovery (risk_event_hedge), 14
process_writeoff (risk_event_hedge), 14
provision_matrix_ecl (core_formulas), 6

reconcile_ledger
(reporting_monitoring), 11
reconcile_overlay (risk_event_hedge), 14
recovery_present_value (core_formulas),
6
reference_data, 3, 5, 9, 18
regulatory_bridge
(reporting_monitoring), 11
reporting_monitoring, 3, 11, 18
risk_event_hedge, 3, 13, 14, 18
run_dataset (workflow), 17
run_dataset(), 2, 10, 13

scenario_ecl (risk_event_hedge), 14
scenario_ecl(), 5, 8, 13
scenario_sensitivity
(reporting_monitoring), 11
select_snapshot (reference_data), 9
split_own_credit
(classification_decisions), 3
stage_movements
(classification_decisions), 3
survival_from_pd (core_formulas), 6

validate_dataset (reference_data), 9
validate_hedge_designation
(risk_event_hedge), 14
validate_overlay (risk_event_hedge), 14

weighted_ecl (core_formulas), 6
workflow, 3, 17