Package 'PRA'

August 28, 2026

Type Package

Title Project Risk Analysis

Version 0.6.0

Description Data analysis for Project Risk Management via the Second Moment Method, Monte Carlo Simulation, Contingency Analysis, Sensitivity Analysis, Earned Value Management, Learning Curves, Bayesian Methods, and more.

Depends R (>= 4.1.0)

Imports mc2d, minpack.lm, stats

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL https://paulgovan.github.io/PRA/, https://github.com/paulgovan/PRA

BugReports https://github.com/paulgovan/PRA/issues

Suggests base64enc, corrplot, devtools, ellmer, ggplot2, igraph, jsonlite, knitr, networkD3, remotes, rmarkdown, scales, mcptools, testthat (>= 3.0.0), withr

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Author Paul Govan [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-1821-8492>)

Maintainer Paul Govan <paul.govan2@gmail.com>

Repository CRAN

Date/Publication 2026-08-28 14:40:17 UTC

Contents

---

ac *Actual Cost (AC).*

Description

Calculates the Actual Cost (AC) of work completed based on the actual costs incurred at each time period.

Usage

```
ac(actual_costs, time_period, cumulative = TRUE)
```

Arguments

actual_costs	Vector of actual costs incurred at each time period. Can be either period costs (cost per period) or cumulative costs depending on the cumulative parameter.
time_period	Current time period.
cumulative	Logical. If TRUE (default), actual_costs are already cumulative and the value at time_period is returned directly. If FALSE, actual_costs are period costs and will be summed up to time_period.

Value

The function returns the Actual Cost (AC) of work completed to date.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

pv, ev, sv, cv, spi, cpi, eac

Examples

```
# Using cumulative costs (default)
cumulative_costs <- c(9000, 27000, 63000, 133000, 233000)
time_period <- 3
ac <- ac(cumulative_costs, time_period)
cat("Actual Cost (AC):", ac, "\n")

# Using period costs
period_costs <- c(9000, 18000, 36000, 70000, 100000)
ac <- ac(period_costs, time_period, cumulative = FALSE)
cat("Actual Cost (AC):", ac, "\n")
```

---

building_project *Mid-Rise Commercial Building Construction Project*

Description

A realistic example project used to illustrate the full PRA workflow: schedule/cost uncertainty, earned value management, Bayesian risk inference, and dependency structure analysis. The project comprises six work packages for a mid-rise commercial building. The structure and parameter ranges are adapted from the illustrative construction-project examples in Damnjanovic and Reinschmidt (2020), the reference text this package operationalizes; the values are representative estimates for a project of this type rather than proprietary data from a specific project.

Usage

```
building_project
```

Format

A named list with the following components:

task_names Character vector of the six work-package names.

task_distributions List of six triangular duration distributions (weeks), each a list with `type = "triangular"` and `a` (optimistic/min), `b` (most likely/mode), and `c` (pessimistic/max). Suitable input to `mcs()` and `sensitivity()`.

cor_mat 6x6 correlation matrix among task durations.

bac Budget at completion (US dollars).

schedule Numeric vector of cumulative planned-value fractions.

actual_costs Numeric vector of per-period actual costs (US dollars).

time_period Integer current reporting period.

actual_per_complete Actual fraction of work complete.

cause_names Character vector of root-cause names for the schedule-delay risk event.

cause_probs Probabilities that each root cause is present.

risks_given_causes P(delay | cause present) for each cause.

risks_given_not_causes P(delay | cause absent) for each cause.

observed_causes Mid-project observation of each cause (1 = occurred, 0 = did not occur, `NA` = not yet assessed).

resource_names Character vector of the six shared resources.

resource_task Resource-task incidence matrix S (resources x tasks) for `parent_dsm()` and `grandparent_dsm()`.

risk_names Character vector of the three structural risks.

risk_resource Risk-resource incidence matrix R (risks x resources) for `grandparent_dsm()`.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data Analytics for Engineering and Construction Project Risk Management. Cham, Switzerland: Springer, 2020. doi:10.1007/9783030142513

Examples

```
# Monte Carlo schedule risk (tasks treated as independent)
sim <- mcs(10000, building_project$task_distributions)
sim$percentiles

# Earned value snapshot at the current period
bp <- building_project
spi(ev(bp$bac, bp$actual_per_complete), pv(bp$bac, bp$schedule, bp$time_period))
```

`contingency` *Contingency Calculation.*

Description

This function calculates the contingency required for a project based on the results of a Monte Carlo simulation. The contingency is determined by the difference between the specified high percentile (phigh) and the base percentile (pbase) of the total project duration distribution.

Usage

```
contingency(sims, phigh = 0.95, pbase = 0.5)
```

Arguments

`sims`	List of results from a Monte Carlo simulation containing the total project duration distribution.
`phigh`	Percentile level for contingency calculation. Default is 0.95 (95th percentile).
`pbase`	Base level for contingency calculation. Default is 0.5 (50th percentile).

Value

The function returns the value of calculated contingency based on the specified percentiles.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Set the number os simulations and the task distributions for a toy project.
num_sims <- 10000
task_dists <- list(
  list(type = "normal", mean = 10, sd = 2), # Task A: Normal distribution
  list(type = "triangular", a = 5, b = 10, c = 15), # Task B: Triangular distribution
  list(type = "uniform", min = 8, max = 12) # Task C: Uniform distribution
)
```

```
# Set the correlation matrix for the correlations between tasks.
cor_mat <- matrix(c(
  1, 0.5, 0.3,
  0.5, 1, 0.4,
  0.3, 0.4, 1
), nrow = 3, byrow = TRUE)

# Run the Monte Carlo simulation.
results <- mcs(num_sims, task_dists, cor_mat)

# Calculate the contingency and print the results.
contingency <- contingency(results, phigh = 0.95, pbase = 0.50)
cat("Contingency based on 95th percentile and 50th percentile:", contingency)

# Without correlation matrix
results_indep <- mcs(num_sims, task_dists)
contingency_indep <- contingency(results_indep,
  phigh = 0.95,
  pbase = 0.50
)
cat("Contingency based on 95th percentile and 50th percentile (
independent tasks):", contingency_indep)

# Build a barplot to visualize the contingency results.
contingency_data <- data.frame(
  Scenario = c("With Correlation", "Independent Tasks"),
  Contingency = c(contingency, contingency_indep)
)
barplot(
  height = contingency_data$Contingency,
  names = contingency_data$Scenario,
  col = c("orange", "purple"),
  horiz = TRUE,
  xlab = "Contingency",
  ylab = "Scenario"
)
title("Contingency Calculation for Project Scenarios")
```

`cor_matrix` *Generate Correlation Matrix from Random Samples.*

Description

This function generates random samples from specified probability distributions and computes the correlation matrix for the generated samples.

Usage

```
cor_matrix(num_samples = 100, num_vars = 5, dists)
```

Arguments

num_samples	The number of samples to generate.
num_vars	The number of distributions to sample. The first num_vars elements of dists are used.
dists	A list describing each distribution. Each element should be a function that generates random samples. The names of the list elements are used to label the rows and columns of the result.

Value

The function returns the correlation matrix for the distributions, with rows and columns named after the distributions they were drawn from. Because the columns are sampled independently, the off-diagonal entries are sampling noise about zero: this generates correctly shaped, positive-definite input for testing and for a near-independent baseline, and is not an estimator of dependence between tasks.

References

Govan, Paul, and Ivan Damnjanovic. "The resource-based view on project risk management." Journal of construction engineering and management 142.9 (2016): 04016034.

Examples

```
# List of probability distributions
dists <- list(
  normal = function(n) rnorm(n, mean = 0, sd = 1),
  uniform = function(n) runif(n, min = 0, max = 1),
  exponential = function(n) rexp(n, rate = 1),
  poisson = function(n) rpois(n, lambda = 1),
  binomial = function(n) rbinom(n, size = 10, prob = 0.5)
)

# Generate correlation matrix
cor_matrix <- cor_matrix(num_samples = 100, num_vars = 5, dists = dists)

# Print correlation matrix
print(cor_matrix)
```

---

cost_pdf *Cost Probability Density.*

Description

This function generates random samples from a mixture model representing the cost 'A' associated with multiple risk events 'R_i'. Each risk event has its own probability, mean, and standard deviation for the cost distribution. The function also accounts for a baseline cost when no risk event occurs.

Usage

```
cost_pdf(
  num_sims,
  risk_probs,
  means_given_risks,
  sds_given_risks,
  base_cost = 0
)
```

Arguments

num_sims	Number of random samples to draw from the mixture model.
risk_probs	A vector of probabilities for each risk event 'R_i'. The risks are independent, so these need not sum to 1.
means_given_risks	A vector of means of the normal distribution for cost 'A' given each risk event 'R_i'.
sds_given_risks	A vector of standard deviations of the normal distribution for cost 'A' given each risk event 'R_i'.
base_cost	The baseline cost given no risk event occurs.

Details

The risk events are independent Bernoulli draws, so any number of them may occur in the same simulation and their probabilities need not sum to one. Each risk that occurs contributes a normal cost on top of base_cost.

Value

A numeric vector of random samples from the mixture model.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Example with three risk events
num_sims <- 1000
risk_probs <- c(0.3, 0.5, 0.2)
means_given_risks <- c(10000, 15000, 5000)
sds_given_risks <- c(2000, 1000, 1000)
base_cost <- 2000
samples <- cost_pdf(
  num_sims = num_sims,
  risk_probs = risk_probs,
  means_given_risks = means_given_risks,
```

```
  sds_given_risks = sds_given_risks,
  base_cost = base_cost
)
hist(samples, breaks = 30, col = "skyblue", main = "Histogram of Cost", xlab = "Cost")
```

`cost_post_pdf` *Posterior Cost Probability Density.*

Description

This function generates random samples from the posterior distribution of the cost 'A' given observations of multiple risk events 'R_i'. Each risk event has its own mean and standard deviation for the cost distribution. The function also accounts for a baseline cost when no risk event occurs.

Usage

```
cost_post_pdf(
  num_sims,
  observed_risks,
  means_given_risks,
  sds_given_risks,
  base_cost = 0,
  risk_probs = NULL
)
```

Arguments

`num_sims`	Number of random samples to draw from the posterior distribution.
`observed_risks`	A vector of observed values for each risk event 'R_i' (1 if observed, 0 if not observed, NA if unobserved).
`means_given_risks`	A vector of means of the normal distribution for cost 'A' given each risk event 'R_i'.
`sds_given_risks`	A vector of standard deviations of the normal distribution for cost 'A' given each risk event 'R_i'.
`base_cost`	The baseline cost given no risk event occurs.
`risk_probs`	Optional vector of prior probabilities for each risk event, used to draw the risks left unobserved (`NA`) in `observed_risks`. If `NULL` (default), unobserved risks are treated as not occurring and a warning is issued.

Details

An observed risk is fixed: a risk observed to have occurred always contributes its cost, and one observed not to have occurred never does. An unobserved risk (`NA`) is drawn from its prior probability when `risk_probs` is supplied, which is the same treatment `risk_post_prob()` gives an unobserved cause. When `risk_probs` is `NULL` there is no prior to draw from, so unobserved risks contribute nothing and the result is a posterior over the observed risks alone; the function warns in that case, because ignoring an unobserved risk understates the cost.

Value

A numeric vector of random samples from the posterior distribution of costs.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Example with three risk events
num_sims <- 1000
observed_risks <- c(1, NA, 1)
means_given_risks <- c(10000, 15000, 5000)
sds_given_risks <- c(2000, 1000, 1000)
base_cost <- 2000
# The second risk is unobserved, so it is drawn from its prior probability.
posterior_samples <- cost_post_pdf(
  num_sims = num_sims,
  observed_risks = observed_risks,
  means_given_risks = means_given_risks,
  sds_given_risks = sds_given_risks,
  base_cost = base_cost,
  risk_probs = c(0.3, 0.5, 0.2)
)
hist(posterior_samples, breaks = 30, col = "skyblue", main = "Posterior Cost PDF", xlab = "Cost")
```

---

cpi *Cost Performance Index (CPI).*

Description

Calculates the Cost Performance Index (CPI) of work completed based on the Earned Value (EV) and Actual Cost (AC).

Usage

```
cpi(ev, ac)
```

Arguments

ev	Earned Value.
ac	Actual Cost.

Value

The function returns the Cost Performance Index (CPI) of work completed.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

pv, ev, ac, sv, cv, spi, eac

Examples

```
# Set the BAC and actual % complete for an example project.
bac <- 100000
actual_per_complete <- 0.35

# Calcualte the EV
ev <- ev(bac, actual_per_complete)

# Set the actual costs and current time period and calculate the AC.
actual_costs <- c(9000, 18000, 36000, 70000, 100000)
time_period <- 3
ac <- ac(actual_costs, time_period)

# Calculate the CPI and print the results.
cpi <- cpi(ev, ac)
cat("Cost Performance Index (CPI):", cpi, "\n")
```

---

cv *Cost Variance (CV).*

Description

Calculates the Cost Variance (CV) of work completed based on the Earned Value (EV) and Actual Cost (AC).

Usage

```
cv(ev, ac)
```

Arguments

ev	Earned Value.
ac	Actual Cost.

Value

The function returns the Cost Variance (CV) of work completed.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

pv, ev, ac, sv, spi, cpi, eac

Examples

```
# Set the BAC and actual % complete for an example project.
bac <- 100000
actual_per_complete <- 0.35

# Calcualte the EV
ev <- ev(bac, actual_per_complete)

# Set the actual costs and current time period and calculate the AC.
actual_costs <- c(9000, 18000, 36000, 70000, 100000)
time_period <- 3
ac <- ac(actual_costs, time_period)

# Calculate the CV and print the results.
cv <- cv(ev, ac)
cat("Cost Variance (CV):", cv, "\n")
```

---

eac *Estimate at Completion (EAC).*

Description

Calculates the Estimate at Completion (EAC) using various methods based on project performance assumptions.

Usage

```
eac(bac, method = "typical", cpi = NULL, ac = NULL, ev = NULL, spi = NULL)
```

Arguments

bac	Budget at Completion (BAC) (total planned budget).
method	The EAC calculation method. One of: • "typical" (default): EAC = BAC / CPI. Assumes future work performed at current cost efficiency. • "atypical": EAC = AC + (BAC - EV). Assumes future work performed at planned rate. • "combined": EAC = AC + (BAC - EV) / (CPI * SPI). Considers both cost and schedule performance.

cpi	Cost Performance Index (CPI). Required for "typical" and "combined" methods.
ac	Actual Cost. Required for "atypical" and "combined" methods.
ev	Earned Value. Required for "atypical" and "combined" methods.
spi	Schedule Performance Index. Required for "combined" method.

Value

The function returns the Estimate at Completion (EAC).

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

pv, ev, ac, sv, cv, spi, cpi, etc, vac, tcpi

Examples

```
# Method 1: Typical - assumes current CPI continues
bac <- 100000
cpi <- 0.83
eac <- eac(bac, cpi = cpi)
cat("EAC (typical):", round(eac, 2), "\n")

# Method 2: Atypical - assumes future work at planned rate
ac <- 63000
ev <- 35000
eac <- eac(bac, method = "atypical", ac = ac, ev = ev)
cat("EAC (atypical):", round(eac, 2), "\n")

# Method 3: Combined - considers both CPI and SPI
spi <- 0.875
eac <- eac(bac, method = "combined", cpi = cpi, ac = ac, ev = ev, spi = spi)
cat("EAC (combined):", round(eac, 2), "\n")
```

---

etc *Estimate to Complete (ETC).*

Description

Calculates the Estimate to Complete (ETC), which is the expected cost to finish the remaining work.

Usage

```
etc(bac, ev, cpi = NULL)
```

Arguments

bac	Budget at Completion (BAC) (total planned budget).
ev	Earned Value.
cpi	Cost Performance Index. If NULL, assumes remaining work will be completed at planned cost (ETC = BAC - EV). If provided, adjusts for current performance (ETC = (BAC - EV) / CPI).

Value

The function returns the Estimate to Complete (ETC).

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

eac, vac, tcpi

Examples

```
bac <- 100000
ev <- 35000
cpi <- 0.83

# ETC assuming remaining work at planned rate
etc <- etc(bac, ev)
cat("ETC (planned rate):", etc, "\n")

# ETC assuming remaining work at current CPI
etc <- etc(bac, ev, cpi)
cat("ETC (current CPI):", round(etc, 2), "\n")
```

---

ev *Earned Value (EV).*

Description

Calculates the Earned Value (EV) of work completed based on the Budget at Completion (BAC) and the actual work completion percentage.

Usage

```
ev(bac, actual_per_complete)
```

Arguments

bac	Budget at Completion (BAC) (total planned budget).
actual_per_complete	Actual work completion percentage.

Value

The function returns the Earned Value (EV) of work completed.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

pv, ac, sv, cv, spi, cpi, eac

Examples

```
# Set the BAC and actual % complete for a toy project.
bac <- 100000
actual_per_complete <- 0.35

# Calculate the EV and print the results.
ev <- ev(bac, actual_per_complete)
cat("Earned Value (EV):", ev, "\n")
```

---

fit_sigmoidal *Fit a Sigmoidal Model to Data.*

Description

This function fits a sigmoidal model (Pearl, Gompertz, or Logistic) to the provided data.

Usage

```
fit_sigmoidal(data, x_col, y_col, model_type)
```

Arguments

data	A data frame containing the time (x_col) and completion (y_col) vectors.
x_col	The name of the time vector.
y_col	The name of the completion vector.
model_type	The name of the sigmoidal model (Pearl, Gompertz, or Logistic).

Value

The function returns a list of results for the sigmoidal model.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Set up a data frame of time and completion percentage data
data <- data.frame(time = 1:10, completion = c(5, 15, 40, 60, 70, 75, 80, 85, 90, 95))

# Fit a logistic model to the data.
fit <- fit_sigmoidal(data, "time", "completion", "logistic")

# Use the model to predict future completion times.
predictions <- predict_sigmoidal(fit, seq(min(data$time), max(data$time),
  length.out = 100
), "logistic")

# Predict with 95% confidence bounds
predictions_ci <- predict_sigmoidal(fit, seq(min(data$time), max(data$time),
  length.out = 100
), "logistic", conf_level = 0.95)
```

---

grandparent_dsm *Risk-based 'Grandparent' Design Structure Matrix (DSM).*

Description

This function computes the Risk-based 'Grandparent' Design Structure Matrix (DSM) from given Resource-Task Matrix 'S' and Risk-Resource Matrix 'R'. The 'Grandparent' DSM scores the risk exposure that each pair of tasks shares through the resource chain. Entry G_{jk} sums, over risks, the product of the risk-to-task path counts $(RS)_{ij}$ and $(RS)_{ik}$, so a shared risk is weighted by the number of common resources carrying it; this reduces to a plain count of shared risks only when RS is binary.

Usage

```
grandparent_dsm(S, R)
```

Arguments

S	Resource-Task Matrix 'S' giving the links (arcs) between resources and tasks. Rows represent resources and columns represent tasks.
R	Risk-Resource Matrix 'R' giving the links (arcs) between risks and resources. Rows represent risks and columns represent resources.

Value

An S3 object of class "dsm" with the following components:

matrix The Risk-based 'Grandparent' DSM giving the shared risk-exposure score for each task pair.

type Character string "grandparent".

n_tasks Number of tasks (columns in S).

n_resources Number of resources (rows in S).

n_risks Number of risks (rows in R).

References

Govan, Paul, and Ivan Damnjanovic. "The resource-based view on project risk management." Journal of construction engineering and management 142.9 (2016): 04016034.

Examples

```
# Set the S and R matrices and print the results.
S <- matrix(c(1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1), nrow = 3, ncol = 4,
            dimnames = list(
              c("Resource-1", "Resource-2", "Resource-3"),
              c("Task-1", "Task-2", "Task-3", "Task-4")
            ))
R <- matrix(c(1, 1, 0, 1, 0, 0), nrow = 2, ncol = 3,
            dimnames = list(
              c("Risk-1", "Risk-2"),
              c("Resource-1", "Resource-2", "Resource-3")
            ))
cat("Resource-Task Matrix (3 resources x 4 tasks):\n")
print(S)
cat("\nRisk-Resource Matrix (2 risks x 3 resources):\n")
print(R)
# Calculate the Risk-based Grandparent Matrix and print the results.
risk_dsm <- grandparent_dsm(S, R)
print(risk_dsm)
```

---

mcs *Monte Carlo Simulation.*

Description

This function performs a Monte Carlo simulation to estimate the total duration of a project based on individual task distributions and an optional correlation matrix.

Usage

```
mcs(num_sims, task_dists, cor_mat = NULL)
```

Arguments

num_sims	The number of simulations to run.
task_dists	A list of lists describing each task distribution with its parameters. Each task distribution should be specified as a list with a "type" field (indicating the distribution type: "normal", "triangular", or "uniform") and the corresponding parameters: for "normal" (mean, sd), for "triangular" (a, b, c), and for "uniform" (min, max). For example: list(list(type = "normal", mean = 10, sd = 2), list(type = "triangular", a = 5, b = 10, c = 15), list(type = "uniform", min = 8, max = 12))
cor_mat	The correlation matrix for the tasks (Optional). If not provided, tasks are assumed to be independent.

Value

The function returns a list of the total mean, variance, standard deviation, and percentiles for the project.

Note

When a correlation matrix is supplied, dependence is induced by Cholesky decomposition. The correlation matrix is factored, independent standard normal scores are multiplied by the Cholesky factor to carry the target correlation, and each column is then returned to its own scale through the normal CDF and that task's inverse CDF. Applying the factor to standard normal scores rather than to the raw task draws is required for the factorization to be valid, since it presumes unit-variance inputs. Every marginal distribution is therefore preserved exactly. The sampled rank correlation matches the target matrix, while the product-moment correlation is approximate: it is attenuated for strongly skewed marginals, an inherent property of the transform. `smm()` remains available when only first and second moments are needed; see the package's design-structure tools for modeling structural dependence directly.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Set the number of simulations and task distributions for a toy project.
num_sims <- 10000
task_dists <- list(
  list(type = "normal", mean = 10, sd = 2), # Task A: Normal distribution
  list(type = "triangular", a = 5, b = 10, c = 15), # Task B: Triangular distribution
  list(type = "uniform", min = 8, max = 12) # Task C: Uniform distribution
)

# Set the correlation matrix for the correlations between tasks.
cor_mat <- matrix(c(
  1, 0.5, 0.3,
  0.5, 1, 0.4,
  0.3, 0.4, 1
```

```
), nrow = 3, byrow = TRUE)

# Run the Monte Carlo sumulation and print the results.
results <- mcs(num_sims, task_dists, cor_mat)
cat("Mean Total Duration:", results$total_mean, "\n")
cat("Variance of Total Variance:", results$total_variance, "\n")
cat("Standard Deviation of Total Duration:", results$total_sd, "\n")
cat("5th Percentile:", results$percentiles[1], "\n")
cat("Median (50th Percentile):", results$percentiles[2], "\n")
cat("95th Percentile:", results$percentiles[3], "\n")
hist(results$total_distribution,
  breaks = 50, main = "Distribution of Total Project Duration",
  xlab = "Total Duration", col = "skyblue", border = "white"
)
legend("topright", legend = c("Total Duration Distribution"), fill = c("skyblue"))
```

`parent_dsm` *Resource-based 'Parent' Design Structure Matrix (DSM).*

Description

This function computes the Resource-based 'Parent' Design Structure Matrix (DSM) from a given Resource-Task Matrix 'S'. The 'Parent' DSM indicates the number of resources shared between each pair of tasks in a project.

Usage

```
parent_dsm(S)
```

Arguments

`S`	Resource-Task Matrix 'S' giving the links (arcs) between resources and tasks. Rows represent resources and columns represent tasks.

Value

An S3 object of class `"dsm"` with the following components:

matrix The Resource-based 'Parent' DSM giving the number of resources shared between each task.

type Character string `"parent"`.

n_tasks Number of tasks (columns in S).

n_resources Number of resources (rows in S).

References

Govan, Paul, and Ivan Damnjanovic. "The resource-based view on project risk management." Journal of construction engineering and management 142.9 (2016): 04016034.

Examples

```
# Set the S matrix for a toy project (3 resources x 4 tasks).
s <- matrix(c(1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1), nrow = 3, ncol = 4,
            dimnames = list(
              c("Resource-1", "Resource-2", "Resource-3"),
              c("Task-1", "Task-2", "Task-3", "Task-4")
            ))
cat("Resource-Task Matrix:\n")
print(s)

# Calculate the Resource-based Parent DSM and print the results.
resource_dsm <- parent_dsm(s)
print(resource_dsm)
```

`plot.dsm` *Plot a DSM heatmap.*

Description

Displays the Design Structure Matrix as a heatmap where color intensity represents the number of shared resources (parent) or risks (grandparent) between task pairs.

Usage

```
## S3 method for class 'dsm'
plot(x, main = NULL, col = NULL, ...)
```

Arguments

`x`	A `dsm` object returned by `parent_dsm()` or `grandparent_dsm()`.
`main`	Optional plot title. If `NULL`, a default title is generated.
`col`	Color palette vector. If `NULL`, uses `grDevices::heat.colors()`.
`...`	Additional arguments passed to `graphics::image()`.

Value

Invisibly returns `x`.

plot_sigmoidal *Plot a Fitted Sigmoidal Model.*

Description

This function creates a base R plot of a fitted sigmoidal model with the original data points, fitted curve, and optional confidence bounds.

Usage

```
plot_sigmoidal(
  fit,
  data,
  x_col,
  y_col,
  model_type,
  conf_level = NULL,
  n_points = 100,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  line_col = "red",
  ci_col = "lightblue",
  pch = 16,
  ...
)
```

Arguments

fit	A fitted sigmoidal model object from fit_sigmoidal.
data	The original data frame used to fit the model.
x_col	The name of the x (time) column in the data.
y_col	The name of the y (completion) column in the data.
model_type	The type of model (pearl, gompertz, or logistic).
conf_level	Optional confidence level for confidence bounds (e.g., 0.95 for 95%). If NULL (default), no confidence bounds are plotted.
n_points	Number of points to use for the fitted curve (default 100).
main	Plot title. If NULL, a default title is generated.
xlab	X-axis label. If NULL, uses x_col.
ylab	Y-axis label. If NULL, uses y_col.
line_col	Color for the fitted curve (default "red").
ci_col	Color for the confidence band (default "lightblue").
pch	Point character for data points (default 16).
...	Additional arguments passed to plot().

Value

Invisibly returns the predictions data frame.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Set up a data frame of time and completion percentage data
data <- data.frame(time = 1:10, completion = c(5, 15, 40, 60, 70, 75, 80, 85, 90, 95))

# Fit a logistic model to the data.
fit <- fit_sigmoidal(data, "time", "completion", "logistic")

# Plot the fitted model
plot_sigmoidal(fit, data, "time", "completion", "logistic")

# Plot with 95% confidence bounds
plot_sigmoidal(fit, data, "time", "completion", "logistic", conf_level = 0.95)

# Customize the plot
plot_sigmoidal(fit, data, "time", "completion", "logistic",
  conf_level = 0.95,
  main = "Project Completion Forecast",
  xlab = "Time (weeks)",
  ylab = "Completion (%)",
  line_col = "blue",
  ci_col = "lightgray"
)
```

---

pra_mcp_server *Start a PRA MCP Server*

Description

Launches an MCP server that exposes all PRA analytical tools via the Model Context Protocol. Once running, Claude Desktop, Claude Code, or any MCP-compatible client can call PRA functions (Monte Carlo simulation, EVM, Bayesian risk, learning curves, DSM, etc.) as native tools.

Usage

```
pra_mcp_server()
```

Details

The server communicates over stdio by default, which is the standard transport for local MCP servers. It reuses the same tool definitions from `pra_tools()`, so any tool updates are automatically reflected. The bundled Agent Skills in `inst/skills/` document when and how an agent should call each tool.

Value

Called for its side effect (starts the MCP server process). Does not return under normal operation.

Examples

```
## Not run:
# Start the server from an R session
pra_mcp_server()

# Or launch directly from the terminal (for use in Claude Code / Desktop):
# Rscript -e "PRA::pra_mcp_server()"

## End(Not run)
```

---

pra_tools *Create PRA Tool Definitions for LLM Agent*

Description

Creates a list of ellmer tool objects that wrap PRA's exported functions for use with an LLM agent. Each tool includes a description that helps the LLM select the appropriate analysis method and properly format parameters.

Usage

```
pra_tools()
```

Details

Tool wrappers handle serialization between the LLM (JSON strings) and R (lists, matrices, data.frames). Complex inputs like task distribution lists and correlation matrices are accepted as JSON strings and deserialized internally.

Large output vectors (e.g., Monte Carlo simulation samples) are summarized to mean, sd, and key percentiles rather than returning the full vector to the LLM. The full results are stored in the package environment for use by downstream tools (e.g., contingency analysis needs the full distribution).

Tool results carry rich HTML display with inline plots via `ellmer::ContentToolResult` for MCP clients that render it, and degrade gracefully to plain text otherwise.

Value

A list of ellmer tool objects.

Examples

```
## Not run:
# The tool set is served to MCP clients by pra_mcp_server().
tools <- pra_tools()
pra_mcp_server()

## End(Not run)
```

`predict_sigmoidal` *Predict a Sigmoidal Function Using Fitted Model.*

Description

This function predicts values using a fitted sigmoidal model (Pearl, Gompertz, or Logistic) over a specified range of time values.

Usage

```
predict_sigmoidal(fit, x_range, model_type, conf_level = NULL)
```

Arguments

`fit`	A list containing the results of a sigmoidal model.
`x_range`	A vector of time values for the prediction.
`model_type`	The type of model (Pearl, Gompertz, or Logistic) for the prediction.
`conf_level`	Optional confidence level for confidence bounds (e.g., 0.95 for 95%). If NULL (default), no confidence bounds are computed.

Value

The function returns a data frame containing the time (x), predicted values (pred), and optionally lower (lwr) and upper (upr) confidence bounds.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Set up a data frame of time and completion percentage data
data <- data.frame(time = 1:10, completion = c(5, 15, 40, 60, 70, 75, 80, 85, 90, 95))

# Fit a logistic model to the data.
fit <- fit_sigmoidal(data, "time", "completion", "logistic")

# Use the model to predict future completion times.
predictions <- predict_sigmoidal(fit, seq(min(data$time), max(data$time),
  length.out = 100
), "logistic")

# Predict with 95% confidence bounds
predictions_ci <- predict_sigmoidal(fit, seq(min(data$time), max(data$time),
  length.out = 100
), "logistic", conf_level = 0.95)
```

`print.dsm` *Print a DSM object.*

Description

Print a DSM object.

Usage

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

Arguments

`x`	A `dsm` object returned by `parent_dsm()` or `grandparent_dsm()`.
`...`	Additional arguments passed to `print.default()`.

Value

Invisibly returns `x`.

print.mcs *Print method for Monte Carlo Simulation results.*

Description

Displays the total mean, variance, standard deviation, and percentiles of the Monte Carlo Simulation results in a readable format.

Usage

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

Arguments

x	An object of class "mcs".
...	Additional arguments (not used).

Value

None. Prints the results to the console.

Examples

```
# Set the number of simulations and task distributions for a toy project.
num_sims <- 10000
task_dists <- list(
  list(type = "normal", mean = 10, sd = 2), # Task A: Normal distribution
  list(type = "triangular", a = 5, b = 10, c = 15), # Task B: Triangular distribution
  list(type = "uniform", min = 8, max = 12) # Task C: Uniform distribution
)

# Set the correlation matrix for the correlations between tasks.
cor_mat <- matrix(c(
  1, 0.5, 0.3,
  0.5, 1, 0.4,
  0.3, 0.4, 1
), nrow = 3, byrow = TRUE)

# Run the Monte Carlo sumulation and print the results.
results <- mcs(num_sims, task_dists, cor_mat)
# print(results)
```

print.pra_sigmoidal_fit *Print method for Sigmoidal Model*

Description

Displays the summary of the fitted sigmoidal model in a readable format.

Usage

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

Arguments

x	An object of class "pra_sigmoidal_fit" as returned by fit_sigmoidal.
...	Additional arguments (not used).

Value

No return value, called for side effects.

Examples

```
# Set up a data frame of time and completion percentage data
data <- data.frame(time = 1:10, completion = c(
  5, 15, 40, 60, 70, 75, 80, 85,
  90, 95
))

# Fit a logistic model to the data.
fit <- fit_sigmoidal(data, "time", "completion", "logistic")
# Print the model summary
print(fit)
```

print.smm *Print method for SMM results.*

Description

This function defines how to print the results of the Second Moment Method (SMM) analysis. It formats the output to display the total mean, variance, and standard deviation in a readable manner.

Usage

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

Arguments

x	An object of class "smm" containing the SMM results.
...	Additional arguments (not used).

Value

None. The function prints the SMM results to the console.

Examples

```
mean <- c(10, 15, 20)
var <- c(4, 9, 16)
cor_mat <- matrix(c(
  1, 0.5, 0.3,
  0.5, 1, 0.4,
  0.3, 0.4, 1
), nrow = 3, byrow = TRUE)
result <- smm(mean, var, cor_mat)
print(result)

# Without correlation matrix (independent tasks)
result <- smm(mean, var)
print(result)
```

---

prob_net *Probabilistic Network of Project Risks.*

Description

This function is part of the probabilistic network module, whose API may still evolve in future versions.

Usage

```
prob_net(nodes, links, distributions = NULL)
```

Arguments

nodes	A data frame containing the nodes of the graph. Must include a column `id` with unique identifiers for each node.
links	A data frame containing the links of the graph. Must include columns `source` and `target` specifying the nodes that form each edge.
distributions	A named list where names correspond to node IDs and values specify discrete probabilities, continuous probability distributions, conditional distributions, or aggregate distributions.

- "discrete": Specifies `values` and `probs`.

- "normal": Specifies `mean` and `sd`.
- "lognormal": Specifies `meanlog` and `sdlog`.
- "uniform": Specifies `min` and `max`.
- "conditional": Specifies a `condition` (a discrete or conditional node) and two distributions (`true_dist`, `false_dist`). The conditional distributions can themselves be discrete or continuous.
- "aggregate": Specifies `nodes` (a list of continuous node IDs to sum).

Details

This function creates a probabilistic network graph representation of project risks that supports discrete and continuous probability distributions.

The links are load-bearing. A conditional node depends on its `condition` and an aggregate node depends on every node it sums, and `prob_net()` requires the edges in `links` to match that declared structure exactly: an edge with no corresponding dependency, or a dependency with no corresponding edge, is an error rather than a silently ignored inconsistency. Nodes must additionally be supplied in a topological order, with every link running from an earlier node to a later one, which is the order `prob_net_sim()` samples in and which also guarantees the graph is acyclic.

Value

A list with:

- `nodes`: The input `nodes` data frame.
- `links`: The input `links` data frame.
- `adjacency_matrix`: A directed matrix with a 1 in `[source, target]` for every edge.
- `distributions`: The input `distributions` list.

Examples

```
nodes <- data.frame(id = c("A", "B", "C", "D"))
links <- data.frame(
  source = c("A", "B", "C"),
  target = c("B", "D", "D")
)
distributions <- list(
  A = list(type = "discrete", values = c(1, 0), probs = c(0.5, 0.5)),
  B = list(
    type = "conditional", condition = "A",
    true_dist  = list(type = "normal", mean = 1, sd = 0.5),
    false_dist = list(type = "lognormal", meanlog = -1, sdlog = 0.5)
  ),
  C = list(type = "uniform", min = 1, max = 5),
  D = list(type = "aggregate", nodes = c("B", "C"))
)
graph <- prob_net(nodes, links, distributions = distributions)
```

prob_net_learn *Perform Bayesian Learning on a Probabilistic Network of Project Risks.*

Description

This function is part of the probabilistic network module, whose API may still evolve in future versions.

Usage

```
prob_net_learn(network, observations = list(), num_samples = 1000)
```

Arguments

network	A prob_net object created by prob_net().
observations	A named list where names are node IDs and values are observed values.
num_samples	Number of samples to simulate for each node (default is 1000).

Details

This function updates a probabilistic network of project risks with observed values for certain nodes and then performs inference to generate posterior distributions for unobserved nodes. The function supports normal, uniform, lognormal, conditional continuous, conditional discrete, discrete, and aggregate (summation) node types.

Conditioning is performed by rejection sampling: the network is simulated forward from its priors (as in prob_net_sim()) and only the draws whose observed nodes equal the supplied values are retained, repeating until num_samples matching draws are collected. Because whole joint draws are filtered, evidence propagates to *upstream* (parent and confounding) nodes as well as downstream ones. This distinguishes observational conditioning ("seeing", the sense of [Pearl 2009]) from intervention ("doing"): only when the observed node is a root cause with no shared ancestry do prob_net_learn() and prob_net_update() induce the same distribution.

Because matches are exact, observations are supported on discrete (or discrete-conditional) nodes; observing a continuous node has probability zero of an exact match and will raise an error. Nodes not listed in observations retain their model distributions. If observations is empty the result is a plain forward simulation.

Value

A data frame with num_samples rows and one column per node containing the simulated posterior samples.

Examples

```
# Define nodes
nodes <- data.frame(
  id = c("A", "B", "C", "D"),
  label = c("Node A", "Node B", "Node C", "Node D"),
  stringsAsFactors = FALSE
)

# Define links
links <- data.frame(
  source = c("A", "B", "C"),
  target = c("C", "D", "D"),
  weight = c(1, 2, 3),
  stringsAsFactors = FALSE
)

# Define distributions for nodes
distributions <- list(
  A = list(type = "discrete", values = c(0, 1), probs = c(0.5, 0.5)),
  B = list(type = "normal", mean = 2, sd = 0.5),
  C = list(
    type = "conditional", condition = "A",
    true_dist = list(type = "normal", mean = 1, sd = 0.5),
    false_dist = list(type = "discrete", values = c(0, 1), probs = c(0.4, 0.6))
  ),
  D = list(type = "aggregate", nodes = c("B", "C"))
)

# Create the network graph
graph <- prob_net(nodes, links, distributions = distributions)

# Perform Bayesian updating with observations
observations <- list(A = 1)
updated_results <- prob_net_learn(graph, observations, num_samples = 1000)
head(updated_results)
```

`prob_net_sim` *Perform Inference on a Probabilistic Network of Project Risks.*

Description

This function is part of the probabilistic network module, whose API may still evolve in future versions.

Usage

```
prob_net_sim(network, num_samples = 1000)
```

Arguments

network	A prob_net object created by prob_net().
num_samples	Number of samples to simulate for each node (default is 1000).

Details

This function performs inference on a probabilistic network of project risks by simulating random samples from the distribution of each node. The function supports normal, uniform, lognormal, discrete, conditional distributions, and aggregate nodes that sum the values of specified continuous nodes.

Aggregate nodes are computed as the sum of values from the specified continuous nodes. Conditional nodes depend on a discrete conditional node; if the condition is true (value = 1), the node follows the true_dist, otherwise it follows the false_dist (value = 0). For discrete distributions, sampling is performed using sample().

Value

A data frame with num_samples rows and one column per node containing the simulated samples.

Examples

```
# Define nodes
nodes <- data.frame(
  id = c("A", "B", "C", "D"),
  label = c("Node A", "Node B", "Node C", "Node D"),
  stringsAsFactors = FALSE
)

# Define links
links <- data.frame(
  source = c("A", "B", "C"),
  target = c("C", "D", "D"),
  weight = c(1, 2, 3),
  stringsAsFactors = FALSE
)

# Define distributions for nodes
distributions <- list(
  A = list(type = "discrete", values = c(0, 1), probs = c(0.5, 0.5)),
  B = list(type = "normal", mean = 2, sd = 0.5),
  C = list(
    type = "conditional", condition = "A",
    true_dist = list(type = "normal", mean = 1, sd = 0.5),
    false_dist = list(type = "lognormal", meanlog = 0, sdlog = 0.2)
  ),
  D = list(type = "aggregate", nodes = c("B", "C"))
)

# Create the network graph
graph <- prob_net(nodes, links, distributions = distributions)
```

```
# Perform inference (simulate 1000 samples)
simulation_results <- prob_net_sim(graph, num_samples = 1000)
head(simulation_results)
```

`prob_net_update` *Update a Probabilistic Network of Project Risks.*

Description

This function is part of the probabilistic network module, whose API may still evolve in future versions.

Usage

```
prob_net_update(
  graph,
  add_links = NULL,
  remove_links = NULL,
  update_distributions = NULL
)
```

Arguments

`graph`	An existing probabilistic network created by `prob_net()`.
`add_links`	Optional. A data frame with columns `source` and `target` to add new links.
`remove_links`	Optional. A data frame with columns `source` and `target` to remove existing links.
`update_distributions`	Optional. A named list of distributions to update. Format follows `prob_net()`.

Details

This function updates an existing probabilistic network by adding or removing dependencies (edges) and updating probability distributions for nodes.

The updated network is re-validated with the same rules `prob_net()` applies, so the edge changes and the distribution changes must agree. Removing the edge into a conditional node without also replacing that node's distribution is an error, which is what makes `remove_links` structurally meaningful: an intervention that severs a dependency has to sever it in both the graph and the distribution list.

Value

An updated `prob_net` object with modified links and/or distributions.

Examples

```
nodes <- data.frame(id = c("A", "B", "C"))
links <- data.frame(source = c("A", "B"), target = c("B", "C"))
distributions <- list(
  A = list(type = "discrete", values = c(1, 0), probs = c(0.5, 0.5)),
  B = list(
    type = "conditional", condition = "A",
    true_dist  = list(type = "normal", mean = 5, sd = 1),
    false_dist = list(type = "normal", mean = 1, sd = 1)
  ),
  C = list(type = "aggregate", nodes = "B")
)
graph <- prob_net(nodes, links, distributions)

# Intervene on B: sever its dependence on A and fix it to the baseline cost.
updated_graph <- prob_net_update(
  graph,
  remove_links = data.frame(source = "A", target = "B"),
  update_distributions = list(B = list(type = "normal", mean = 1, sd = 1))
)
```

---

pv *Planned Value (PV).*

Description

Calculates the Planned Value (PV) of work completed based on the Budget at Completion (BAC) and the planned schedule.

Usage

```
pv(bac, schedule, time_period)
```

Arguments

bac	Budget at Completion (BAC) (total planned budget).
schedule	Vector of planned work completion (in terms of percentage) at each time period.
time_period	Current time period.

Value

The function returns the Planned Value (PV) of work completed.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

ev, ac, sv, cv, spi, cpi, eac

Examples

```
# Set the BAC, schedule, and current time period for a toy project.
bac <- 100000
schedule <- c(0.1, 0.2, 0.4, 0.7, 1.0)
time_period <- 3

# Calculate the PV and print the results.
pv <- pv(bac, schedule, time_period)
cat("Planned Value (PV):", pv, "\n")
```

---

risk_post_prob *Posterior Risk Probability.*

Description

This function calculates the posterior probability of a risk event 'R' occurring based on observations of multiple root causes and their associated conditional probabilities.

Usage

```
risk_post_prob(
  cause_probs,
  risks_given_causes,
  risks_given_not_causes,
  observed_causes
)
```

Arguments

cause_probs	A vector of prior probabilities for each root cause 'C_i'.
risks_given_causes	A vector of conditional probabilities of the risk event 'R' given each cause 'C_i'.
risks_given_not_causes	A vector of conditional probabilities of the risk event 'R' given not each cause 'C_i'.
observed_causes	A vector of observed values for each cause 'C_i' (1 if observed, 0 if not observed, NA if unobserved).

Details

Observed causes are fixed to their conditional probability ($P(R \mid C_i)$ if present, $P(R \mid \bar{C}_i)$ if absent) while unobserved causes (NA) keep their marginal contribution. The causes are then combined with the same noisy-OR as `risk_prob()`, so the posterior is on the same scale as the prior and observing an aggravating cause raises it.

Value

A numeric value for the posterior probability of the risk event given the observed causes.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
cause_probs <- c(0.3, 0.2)
risks_given_causes <- c(0.8, 0.6)
risks_given_not_causes <- c(0.2, 0.4)
observed_causes <- c(1, NA)
risk_post_prob <- risk_post_prob(
  cause_probs, risks_given_causes,
  risks_given_not_causes, observed_causes
)
print(risk_post_prob)
```

`risk_prob` *Bayesian Inference for Risk Probability.*

Description

This function calculates the overall probability of a risk event 'R' occurring based on the probabilities of multiple root causes and their associated conditional probabilities.

Usage

```
risk_prob(cause_probs, risks_given_causes, risks_given_not_causes)
```

Arguments

`cause_probs`	A vector of probabilities for each root cause 'C_i'.
`risks_given_causes`	A vector of conditional probabilities of the risk event 'R' given each cause 'C_i'.
`risks_given_not_causes`	A vector of conditional probabilities of the risk event 'R' given not each cause 'C_i'.

Details

Each cause contributes a marginal probability $P(R \mid C_i)P(C_i) + P(R \mid \bar{C}_i)P(\bar{C}_i)$ via the law of total probability. Independent causes are combined with a noisy-OR — the probability that the risk event is triggered by at least one cause — so the result always lies in $[0, 1]$ and reduces to the single-cause marginal when there is one cause.

Value

The function returns a numeric value for the probability of risk event 'R'.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
cause_probs <- c(0.3, 0.2)
risks_given_causes <- c(0.8, 0.6)
risks_given_not_causes <- c(0.2, 0.4)
risk_prob_value <- risk_prob(cause_probs, risks_given_causes, risks_given_not_causes)
print(risk_prob_value)
```

---

sensitivity *Sensitivity Analysis.*

Description

This function performs sensitivity analysis on a project with multiple tasks, each having its own cost distribution. It calculates the sensitivity of the variance in total project cost with respect to the variance in each task's cost. It can also account for correlations between task costs if a correlation matrix is provided.

Usage

```
sensitivity(task_dists, cor_mat = NULL)
```

Arguments

task_dists	A list of lists describing each task distribution. Each inner list should contain the type of distribution and its parameters. Supported distributions are "normal", "triangular", and "uniform".
cor_mat	The correlation matrix for the tasks (Optional). If provided, it should be a square matrix with dimensions equal to the number of tasks. If not provided, tasks are assumed to be independent.

Value

The function returns a vector of sensitivity results with respect to each task. Each element is that task's contribution (own variance plus its covariance with all other tasks) to total project variance, expressed as a proportion of total variance.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Set the task distributions for a toy project.
task_dists <- list(
  list(type = "normal", mean = 10, sd = 2), # Task A: Normal distribution
  list(type = "triangular", a = 5, b = 15, c = 10), # Task B: Triangular distribution
  list(type = "uniform", min = 8, max = 12) # Task C: Uniform distribution
)

# Set the correlation matrix between the tasks.
cor_mat <- matrix(c(
  1, 0.5, 0.3,
  0.5, 1, 0.4,
  0.3, 0.4, 1
), nrow = 3, byrow = TRUE)

# Calculate the sensitivity of each task and print the results
sensitivity_results <- sensitivity(task_dists, cor_mat)
print(sensitivity_results)

# Build a vertical barchart and display the results.
data <- data.frame(
  Tasks = c("A", "B", "C"),
  Sensitivity = sensitivity_results
)
barplot(
  height = data$Sensitivity, names = data$Tasks, col = "skyblue",
  horiz = TRUE, xlab = "Sensitivity", ylab = "Tasks"
)
title("Sensitivity Analysis of Project Tasks")

# Without correlation matrix
sensitivity_results_indep <- sensitivity(task_dists)
print(sensitivity_results_indep)

# Build a vertical barchart and display the results.
data_indep <- data.frame(
  Tasks = c("A", "B", "C"),
  Sensitivity = sensitivity_results_indep
)
barplot(
  height = data_indep$Sensitivity, names = data_indep$Tasks,
```

```
  col = "lightgreen",
  horiz = TRUE, xlab = "Sensitivity", ylab = "Tasks"
)
title("Sensitivity Analysis of Project Tasks (Independent)")
```

`smm` *Second Moment Method Analysis.*

Description

This function performs the Second Moment Method (SMM) analysis to estimate the total mean, variance, and standard deviation of a project based on individual task means, variances, and an optional correlation matrix.

Usage

```
smm(mean, var, cor_mat = NULL)
```

Arguments

`mean`	The mean vector.
`var`	The variance vector.
`cor_mat`	The correlation matrix (optional). If not provided, tasks are assumed to be independent.

Value

The function returns a list of the total mean, variance, and standard deviation for the project.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

Examples

```
# Set the mean vector, variance vector, and correlation matrix for a toy project.
mean <- c(10, 15, 20)
var <- c(4, 9, 16)
cor_mat <- matrix(c(
  1, 0.5, 0.3,
  0.5, 1, 0.4,
  0.3, 0.4, 1
), nrow = 3, byrow = TRUE)

# Use the Second Moment Method to estimate the results for the project.
result <- smm(mean, var, cor_mat)
```

```
print(result)

# Without correlation matrix (independent tasks)
result <- smm(mean, var)
print(result)

# When certain tasks are discrete and others are continuous, the SMM can still
# be applied as long as the variance values accurately reflect the variability of each task.

discrete_mean <- c(5, 10)
discrete_var <- c(0, 0)
continuous_mean <- c(15, 20)
continuous_var <- c(4, 5)
mean <- c(discrete_mean, continuous_mean)
var <- c(discrete_var, continuous_var)
cor_mat <- matrix(c(
  1, 0, 0.2, 0.3,
  0, 1, 0.1, 0.2,
  0.2, 0.1, 1, 0.4,
  0.3, 0.2, 0.4,
  1
), nrow = 4, byrow = TRUE)
result <- smm(mean, var, cor_mat)
print(result)
```

---

spi *Schedule Performance Index (SPI).*

Description

Calculates the Schedule Performance Index (SPI) of work completed based on the Earned Value (EV) and Planned Value (PV).

Usage

```
spi(ev, pv)
```

Arguments

ev	Earned Value.
pv	Planned Value.

Value

The function returns the Schedule Performance Index (SPI) of work completed.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

pv, ev, ac, sv, cv, cpi, eac

Examples

```
# Set the BAC, schedule, and current time period for an example project.
bac <- 100000
schedule <- c(0.1, 0.2, 0.4, 0.7, 1.0)
time_period <- 3

# Calculate the PV.
pv <- pv(bac, schedule, time_period)

# Set the actual % complete and calculate the EV.
actual_per_complete <- 0.35
ev <- ev(bac, actual_per_complete)

# Calculate the SPI and print the results.
spi <- spi(ev, pv)
cat("Schedule Performance Index (SPI):", spi, "\n")
```

---

sv *Schedule Variance (SV).*

Description

Calculates the Schedule Variance (SV) of work completed based on the Earned Value (EV) and Planned Value (PV).

Usage

```
sv(ev, pv)
```

Arguments

ev	Earned Value.
pv	Planned Value.

Value

The function returns the Schedule Variance (SV) of work completed.

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

pv, ev, ac, cv, spi, cpi, eac

Examples

```
# Set the BAC, schedule, and current time period for an example project.
bac <- 100000
schedule <- c(0.1, 0.2, 0.4, 0.7, 1.0)
time_period <- 3

# Calculate the PV.
pv <- pv(bac, schedule, time_period)

# Set the actual % complete and calculate the EV.
actual_per_complete <- 0.35
ev <- ev(bac, actual_per_complete)

# Calculate the SV and print the results.
sv <- sv(ev, pv)
cat("Schedule Variance (SV):", sv, "\n")
```

---

tcpi *To-Complete Performance Index (TCPI).*

Description

Calculates the To-Complete Performance Index (TCPI), which indicates the cost performance required on remaining work to meet a target (BAC or EAC). TCPI > 1 means efficiency must improve; TCPI < 1 means efficiency can decrease.

Usage

```
tcpi(bac, ev, ac, target = "bac", eac = NULL)
```

Arguments

bac	Budget at Completion (BAC) (total planned budget).
ev	Earned Value.
ac	Actual Cost.
target	The target to calculate TCPI against. Either "bac" (default) to meet original budget, or "eac" to meet revised estimate. If "eac", the eac parameter must be provided.
eac	Estimate at Completion. Required when target = "eac".

Value

The function returns the To-Complete Performance Index (TCPI).

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

eac, etc, vac, cpi

Examples

```
bac <- 100000
ev <- 35000
ac <- 63000

# TCPI to complete within original budget
tcpi_bac <- tcpi(bac, ev, ac)
cat("TCPI (to meet BAC):", round(tcpi_bac, 2), "\n")

# TCPI to complete within revised estimate
eac <- 120482
tcpi_eac <- tcpi(bac, ev, ac, target = "eac", eac = eac)
cat("TCPI (to meet EAC):", round(tcpi_eac, 2), "\n")
```

---

vac *Variance at Completion (VAC).*

Description

Calculates the Variance at Completion (VAC), which is the difference between the budget and the expected final cost. Positive VAC indicates under budget, negative indicates over budget.

Usage

```
vac(bac, eac)
```

Arguments

bac	Budget at Completion (BAC) (total planned budget).
eac	Estimate at Completion.

Value

The function returns the Variance at Completion (VAC).

References

Damnjanovic, Ivan, and Kenneth Reinschmidt. Data analytics for engineering and construction project risk management. No. 172534. Cham, Switzerland: Springer, 2020.

See Also

eac, etc, tcpi

Examples

```
bac <- 100000
eac <- 120482 # From EAC calculation

vac <- vac(bac, eac)
cat("Variance at Completion (VAC):", round(vac, 2), "\n")
cat("Project is expected to be", abs(round(vac, 2)), ifelse(vac < 0, "over", "under"), "budget\n")
```

Index