Package ‘collector’

February 18, 2020

Title  Quantified Risk Assessment Data Collection
Version  0.1.3
Description  An open source process for collecting quantified data inputs from subject matter experts. Intended for feeding into an OpenFAIR analysis <https://www2.opengroup.org/ogsys/catalog/C13K> using a tool such as 'evaluator' <https://evaluator.tidyrisk.org>.
Depends  R (>= 3.4.0)
License  MIT + file LICENSE
Encoding  UTF-8
LazyData  true
RoxygenNote  7.0.2
Imports  EnvStats, dplyr, evaluator (>= 0.4.0), flextable, ggplot2, ggpubr, magrittr, purrr, quanteda, readr, rlang, rmarkdown, stringr, tibble, tidyr (>= 1.0.0), officer, xaringan
Suggests  spelling, testthat, covr, knitr
SystemRequirements  pandoc
URL  https://collector.tidyrisk.org
BugReports  https://github.com/davidski/collector/issues
Language  en-US
VignetteBuilder  knitr
NeedsCompilation  no
Author  David Severski [aut, cre] (<https://orcid.org/0000-0001-7867-0459>)
Maintainer  David Severski <davidski@deadheaven.com>
Repository  CRAN
Date/Publication  2020-02-18 00:10:02 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>calibration_questions</td>
<td>3</td>
</tr>
<tr>
<td>check_readability</td>
<td>3</td>
</tr>
<tr>
<td>clean_answers</td>
<td>4</td>
</tr>
<tr>
<td>collector</td>
<td>5</td>
</tr>
<tr>
<td>combine_capability_parameters</td>
<td>5</td>
</tr>
<tr>
<td>combine_lognorm</td>
<td>6</td>
</tr>
<tr>
<td>combine_lognorm_trunc</td>
<td>6</td>
</tr>
<tr>
<td>combine_norm</td>
<td>7</td>
</tr>
<tr>
<td>combine_scenario_parameters</td>
<td>8</td>
</tr>
<tr>
<td>derive_controls</td>
<td>8</td>
</tr>
<tr>
<td>enforce_tidyrisk_question_set</td>
<td>9</td>
</tr>
<tr>
<td>enforce_tidyrisk_response_set</td>
<td>10</td>
</tr>
<tr>
<td>fit_capabilities</td>
<td>10</td>
</tr>
<tr>
<td>fit_capabilities_geomean</td>
<td>11</td>
</tr>
<tr>
<td>fit_lognorm</td>
<td>12</td>
</tr>
<tr>
<td>fit_lognorm_trunc</td>
<td>12</td>
</tr>
<tr>
<td>fit_norm_trunc</td>
<td>13</td>
</tr>
<tr>
<td>fit_pois</td>
<td>14</td>
</tr>
<tr>
<td>fit_scenarios</td>
<td>15</td>
</tr>
<tr>
<td>fit_scenarios_geomean</td>
<td>16</td>
</tr>
<tr>
<td>fit_threat_communities</td>
<td>16</td>
</tr>
<tr>
<td>generate_cost_function</td>
<td>17</td>
</tr>
<tr>
<td>generate_weights</td>
<td>18</td>
</tr>
<tr>
<td>get_smes_domains</td>
<td>18</td>
</tr>
<tr>
<td>is_tidyrisk_question_set</td>
<td>19</td>
</tr>
<tr>
<td>is_tidyrisk_response_set</td>
<td>19</td>
</tr>
<tr>
<td>lognormal_to_normal</td>
<td>20</td>
</tr>
<tr>
<td>make_handouts</td>
<td>21</td>
</tr>
<tr>
<td>make_scorecard</td>
<td>21</td>
</tr>
<tr>
<td>make_slides</td>
<td>22</td>
</tr>
<tr>
<td>mc_calibration_answers</td>
<td>23</td>
</tr>
<tr>
<td>mc_capabilities</td>
<td>24</td>
</tr>
<tr>
<td>mc_capability_answers</td>
<td>24</td>
</tr>
<tr>
<td>mc_capability_parameters_fitted</td>
<td>25</td>
</tr>
<tr>
<td>mc_domains</td>
<td>25</td>
</tr>
<tr>
<td>mc_scenarios</td>
<td>26</td>
</tr>
<tr>
<td>mc_scenario_answers</td>
<td>27</td>
</tr>
<tr>
<td>mc_scenario_parameters_fitted</td>
<td>27</td>
</tr>
<tr>
<td>mc_sme_top_domains</td>
<td>28</td>
</tr>
<tr>
<td>mc_threat_communities</td>
<td>29</td>
</tr>
<tr>
<td>mc_threat_parameters_fitted</td>
<td>29</td>
</tr>
<tr>
<td>normal_to_lognormal</td>
<td>30</td>
</tr>
<tr>
<td>prepare_data</td>
<td>31</td>
</tr>
<tr>
<td>read_questions</td>
<td>32</td>
</tr>
<tr>
<td>read_responses</td>
<td>33</td>
</tr>
<tr>
<td>tidyrisk_question_set</td>
<td>34</td>
</tr>
</tbody>
</table>
**calibration_questions**

A dataset of reference trivia questions for calibrating SMEs.

**Usage**

`calibration_questions`

**Format**

A data frame with 27 rows and 3 variables:

- **question**: text of the calibration question
- **answer**: answer text to the calibration question
- **calibration_id**: unique identifier for the calibration question

**Source**

Common trivia questions drawn from a variety of open source web resources.

---

**check_readability**

Check the readability of scenario text

**Description**

Calculate the Flesch-Kincaid score for each scenario and return that score along with the scenario ID and domain as a tidy dataframe.

**Usage**

`check_readability(x)`

**Arguments**

- **x**: A tidyrisk_question_set object

**Value**

A dataframe of the scenario id, domain id, and the Flesch-Kincaid readability score for the scenario text.
clean_answers

Examples

```r
## Not run:
questions <- read_questions()
check_readability(questions)

## End(Not run)
```

---

clean_answers | Clean extreme answers

Description

You may wish to apply some sanity checking bounds on the responses from subject matter experts. This function applies a set of predefined transformations to the scenario and capability responses. Review these assumptions carefully before using them in your own analysis.

Usage

```r
clean_answers(scenario_answers, capability_answers)
```

Arguments

- `scenario_answers`
  - Scenario answers dataframe.
- `capability_answers`
  - Capability answers dataframe.

Details

Make the following assumptions/modifications

- minimum capacity is 5% (we’ve thought about it - 90% CI)
- maximum capacity is 95% (we’re just about the best - 90% CI)
- minimum loss is 1000 dollars (both low and high)
- scale all impact into thousands of dollars (make normal decomposition easier, and is in line of the scale of a strategic analysis)
- set a minimum frequency of once per 10 years (0.1)

Value

A list of modified scenarios and capabilities.

Examples

```r
data(mc_capability_answers)
data(mc_scenario_answers)
clean_answers(mc_scenario_answers, mc_capability_answers)
```
**Description**

Quantified Information Risk Assessment Data Collection

**Details**

See the online documentation located at [https://evaluator.tidyrisk.org/](https://evaluator.tidyrisk.org/)

---

**combine_capability_parameters**

*Combine multiple SME distributions into a single unified view*

**Description**

Given a dataframe with multiple SME fitted distributions for a single capability, apply weighting for opinion pooling, and construct a final combined distribution for each OpenFAIR scenario parameter.

**Usage**

```
combine_capability_parameters(capability_parameters)
```

**Arguments**

- `capability_parameters`
  
  Fitted individual parameters for capabilities.

**Value**

A dataframe.

**Examples**

```
NULL
```
combine_lognorm_trunc

Weight a set of lognormal parameters into a single distribution

Description
Weight a set of lognormal parameters into a single distribution

Usage
combine_lognorm_trunc(dat)

Arguments
dat  Dataframe of meanlog, sdlog, min, max, and sdlog.

Value
A dataframe.

See Also
Other distribution fitting functions: combine_lognorm_trunc(), combine_norm(), fit_capabilities_geomean(), fit_capabilities(), fit_lognorm_trunc(), fit_lognorm(), fit_norm_trunc(), fit_pois(), fit_scenarios_geomean(), fit_scenarios(), fit_threat_communities(), generate_cost_function(), lognormal_to_normal(), normal_to_lognormal()

Examples
dat <- data.frame(meanlog = c(1, 1.5),
                  sdlog = c(1, 2),
                  weight = c(2, 1))
combine_lognorm_trunc(dat)

combine_lognorm

Weight a set of lognormal parameters into a single distribution

Description
Weight a set of lognormal parameters into a single distribution

Usage
combine_lognorm(dat)

Arguments
dat  A dataframe.

Value
A dataframe.

See Also
Other distribution fitting functions: combine_lognorm_trunc(), combine_norm(), fit_capabilities_geomean(), fit_capabilities(), fit_lognorm_trunc(), fit_lognorm(), fit_norm_trunc(), fit_pois(), fit_scenarios_geomean(), fit_scenarios(), fit_threat_communities(), generate_cost_function(), lognormal_to_normal(), normal_to_lognormal()

Examples
dat <- data.frame(meanlog = c(1, 1.5),
                  sdlog = c(1, 2),
                  weight = c(2, 1))
combine_lognorm(dat)
\texttt{combine\_norm}

\textbf{Value}

A dataframe.

\textbf{See Also}

Other distribution fitting functions: \texttt{combine\_lognorm()}, \texttt{combine\_norm()}, \texttt{fit\_capabilities\_geomean()}, \texttt{fit\_capabilities()}, \texttt{fit\_lognorm\_trunc()}, \texttt{fit\_lognorm()}, \texttt{fit\_norm\_trunc()}, \texttt{fit\_pois()}, \texttt{fit\_scenarios\_geomean()}, \texttt{fit\_scenarios()}, \texttt{fit\_threat\_communities()}, \texttt{generate\_cost\_function()}, \texttt{lognormal\_to\_normal()}, \texttt{normal\_to\_lognormal()}

\textbf{Examples}

\begin{verbatim}
dat <- data.frame(meanlog = c(1, 1.5),
                 sdlog = c(1, 2),
                 min = 0,
                 max = Inf,
                 weight = c(2, 1))
combine_lognorm\_trunc(dat)
\end{verbatim}

\begin{verbatim}
  combine\_norm(dat)
\end{verbatim}

\textbf{Description}

Given a set of arbitrary parameters that includes at least a weight column, take a weighted average of all the other parameters.

\textbf{Usage}

\begin{verbatim}
  combine\_norm(dat)
\end{verbatim}

\textbf{Arguments}

\begin{verbatim}
  dat          Dataframe of mean, sd and weights.
\end{verbatim}

\textbf{Value}

A dataframe.

\textbf{See Also}

Other distribution fitting functions: \texttt{combine\_lognorm\_trunc()}, \texttt{combine\_lognorm()}, \texttt{fit\_capabilities\_geomean()}, \texttt{fit\_capabilities()}, \texttt{fit\_lognorm\_trunc()}, \texttt{fit\_lognorm()}, \texttt{fit\_norm\_trunc()}, \texttt{fit\_pois()}, \texttt{fit\_scenarios\_geomean()}, \texttt{fit\_scenarios()}, \texttt{fit\_threat\_communities()}, \texttt{generate\_cost\_function()}, \texttt{lognormal\_to\_normal()}, \texttt{normal\_to\_lognormal()}
Examples

```r
dat <- data.frame(mean = c(10, 20, 30),
                  sd = c(4, 5, 10),
                  weight = c(2, 1, 2))
combine_norm(dat)
```

---

**combine_scenario_parameters**

*Combine multiple SME distributions into a single unified view*

**Description**

Given a dataframe with multiple SME fitted distributions for a single scenario, decompose the lognormal distribution into normal parameters, apply weighting for opinion pooling, and construct a final combined distribution for each OpenFAIR scenario factor.

**Usage**

```r
combine_scenario_parameters(scenario_parameters)
```

**Arguments**

- `scenario_parameters`
  - Fitted scenario factors for individual SMEs.

**Value**

A dataframe.

**Examples**

```r
NULL
```

---

**derive_controls**

*Generate the quantified capability parameters for a scenario*

**Description**

Based on the evaluator::derive_controls function

**Usage**

```r
derive_controls(capability_ids, capability_parameters)
```
enforce_tidyrisk_question_set

Arguments

capability_ids  Comma-delimited list of capability ids
capability_parameters  Dataframe of fitted and combined capability parameters

Details

Creates the difficulty parameters (embedded list) for quantitative parameters.

Value

A list.

See Also

evaluator::derive_controls

Examples

NULL

enforce_tidyrisk_question_set

Validate that the parameter passed is a tidyrisk_question_set object

Description

Validate that the parameter passed is a tidyrisk_question_set object

Usage

enforce_tidyrisk_question_set(x)

Arguments

x  An object

Examples

NULL
fit_capabilities

**Description**

Fit SME capability estimates to distribution parameters

**Usage**

```r
fit_capabilities(responses)
```

**Arguments**

- `responses` A `tidyrisk_response_set` object

**Value**

A dataframe.

**See Also**

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`, `combine_norm()`, `fit_capabilities_geomean()`, `fit_lognorm_trunc()`, `fit_lognorm()`, `fit_norm_trunc()`, `fit_pois()`, `fit_scenarios_geomean()`, `fit_scenarios()`, `fit_threat_communities()`, `generate_cost_function()`, `lognormal_to_normal()`, `normal_to_lognormal()`
fit_capabilities_geomean

Examples

NULL

---

fit_capabilities_geomean

*Fit capability parameters via a geometric mean*

Description

Fit capability parameters via a geometric mean

Usage

`fit_capabilities_geomean(capabilities_answers)`

Arguments

capabilities_answers

Answers dataframe.

Value

A dataframe.

See Also

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`, `combine_norm()`, `fit_capabilities()`, `fit_lognorm_trunc()`, `fit_lognorm()`, `fit_norm_trunc()`, `fit_pois()`, `fit_scenarios_geomean()`, `fit_scenarios()`, `fit_threat_communities()`, `generate_cost_function()`, `lognormal_to_normal()`, `normal_to_lognormal()`

Examples

```r
data(mc_capability_answers)
fit_capabilities_geomean(mc_capability_answers)
```
**fit_lognorm**

*Find parameters that fit quantile values of an unknown lognormal distribution*

**Description**

With a 5th and 95th quantile point estimates, fit a lognormal distribution, returning the parameters of the distribution.

**Usage**

```
fit_lognorm(low, high)
```

**Arguments**

- `low` 5th quantile.
- `high` 95th quantile.

**Value**

A dataframe.

**See Also**

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`, `combine_norm()`, `fit_capabilities_geomean()`, `fit_capabilities()`, `fit_lognorm_trunc()`, `fit_norm_trunc()`, `fit_pois()`, `fit_scenarios_geomean()`, `fit_scenarios()`, `fit_threat_communities()`, `generate_cost_function()`, `lognormal_to_normal()`, `normal_to_lognormal()`

**Examples**

```
fit_lognorm(low = .20, high = .50)
```

---

**fit_lognorm_trunc**

*Find parameters that fit quantile values of an unknown truncated lognormal distribution*

**Description**

With a 5th and 95th quantile point estimates and optional lower and upper bounds, fit a lognormal distribution, returning the parameters of the distribution.

**Usage**

```
fit_lognorm_trunc(low, high, min = 0, max = Inf)
```
Arguments

- **low**: 5th quantile.
- **high**: 95th quantile.
- **min**: lower bound of support.
- **max**: upper bound of support.

Value

A dataframe.

See Also

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`, `combine_norm()`, `fit_capabilities_geomean()`, `fit_capabilities()`, `fit_lognorm()`, `fit_norm_trunc()`, `fit_pois()`, `fit_scenarios_geomean()`, `fit_scenarios()`, `fit_threat_communities()`, `generate_cost_function()`, `lognormal_to_normal()`, `normal_to_lognormal()`

Examples

```r
fit_lognorm_trunc(low = 10, high = 50, min = 0, max = 100)
```

Description

With a 5th and 95th quantile point estimates and optional lower and upper bounds, fit a truncated normal distribution, returning the parameters of the distribution.

Usage

```r
fit_norm_trunc(low, high, min = 0, max = Inf)
```

Arguments

- **low**: 5th quantile.
- **high**: 95th quantile.
- **min**: Lower bound of support.
- **max**: Upper bound of support.

Value

Dataframe.
See Also

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`, `combine_norm()`, `fit_capabilities_geomean()`, `fit_capabilities()`, `fit_lognorm_trunc()`, `fit_lognorm()`, `fit_pois()`, `fit_scenarios_geomean()`, `fit_scenarios()`, `fit_threat_communities()`, `generate_cost_function()`, `lognormal_to_normal()`, `normal_to_lognormal()`

Examples

```r
fit_pois(low = 10, high = 50)
```

---

**Description**

With a 5th and 95th quantile point estimates and optional lower and upper bounds, fit a poisson distribution, returning the parameters of the distribution.

**Usage**

```r
fit_pois(low, high)
```

**Arguments**

- `low` 5th quantile.
- `high` 95th quantile.

**Value**

A dataframe.

**See Also**

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`, `combine_norm()`, `fit_capabilities_geomean()`, `fit_capabilities()`, `fit_lognorm_trunc()`, `fit_lognorm()`, `fit_norm_trunc()`, `fit_scenarios_geomean()`, `fit_scenarios()`, `fit_threat_communities()`, `generate_cost_function()`, `lognormal_to_normal()`, `normal_to_lognormal()`

**Examples**

```r
fit_pois(low = 10, high = 50)
```
**fit_scenarios**  

*Fit SME scenario estimates to distribution parameters*

**Description**

Given a set of subject matter expert estimates for the 5th and 95th quantiles of impact and frequency of contact for events, calculate the distribution parameters for TEF and LM. Use a truncated lognormal distribution for LM (losses cannot be infinite in size) and for the TEF.

**Usage**

```r
fit_scenarios(
  responses,
  maximum_impact = Inf,
  maximum_impact_factor = 10,
  maximum_frequency_factor = 10
)
```

**Arguments**

- `responses`  
  A `tidyrisk_response_set` object.
- `maximum_impact`  
  The absolute maximum potential impact of any single loss event.
- `maximum_impact_factor`  
  Maximum impact factor - scaling factor of a SME’s 95 percent maximum loss to limit the impact of any single event.
- `maximum_frequency_factor`  
  Maximum frequency factor - scaling factor at which to limit frequency of events.

**Value**

A dataframe.

**See Also**

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`, `combine_norm()`, `fit_capabilities_geomean()`, `fit_capabilities()`, `fit_lognorm_trunc()`, `fit_lognorm()`, `fit_norm_trunc()`, `fit_pois()`, `fit_scenarios_geomean()`, `fit_threat_communities()`, `generate_cost_function()`, `lognormal_to_normal()`, `normal_to_lognormal()`

**Examples**

```r
NULL
```
fit_scenarios_geomean  
Fit scenario parameters by applying a geometric mean

Description

Fit scenario parameters by applying a geometric mean

Usage

fit_scenarios_geomean(scenario_answers)

Arguments

scenario_answers

Scenario answers dataframe.

Value

A dataframe.

See Also

Other distribution fitting functions: combine_lognorm_trunc(), combine_lognorm(), combine_norm(), fit_capabilities_geomean(), fit_capabilities(), fit_lognorm_trunc(), fit_lognorm(), fit_norm_trunc(), fit_pois(), fit_scenarios(), fit_threat_communities(), generate_cost_function(), lognormal_to_normal(), normal_to_lognormal()

Examples

data(mc_scenario_answers)
fitted <- fit_scenarios_geomean(mc_scenario_answers)

fit_threat_communities  
Fit each of the threat communities to a distribution

Description

Fit each of the threat communities to a distribution

Usage

fit_threat_communities(threat_communities)
generate_cost_function

Arguments

threat_communities
    Dataframe of threat communities.

Value

A dataframe.

See Also

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`,
`combine_norm()`,
`fit_capabilities_geomean()`, `fit_capabilities()`,
`fit_lognorm_trunc()`,
`fit_lognorm()`,
`fit_norm_trunc()`,
`fit_pois()`,
`fit_scenarios_geomean()`,
`fit_scenarios()`,
`generate_cost_function()`,
`lognormal_to_normal()`,
`normal_to_lognormal()`

Examples

```r
data(mc_threat_communities)
fit_threat_communities(mc_threat_communities)
```

---

generate_cost_function

Generate a sum of squares cost function for optimization

Description

This is an internal helper function that generates a sum of squares cost function for any given r*
function (e.g. rnorm, rlognorm). The resulting function is intended to be used by an optim call for
fitting quantiles to distribution parameters.

Usage

generate_cost_function(func)

Arguments

func  A distribution function.

Value

A function.

See Also

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`,
`combine_norm()`,
`fit_capabilities_geomean()`, `fit_capabilities()`,
`fit_lognorm_trunc()`, `fit_lognorm()`,
`fit_norm_trunc()`, `fit_pois()`,
`fit_scenarios_geomean()`, `fit_scenarios()`,
`generate_cost_function()`,
`lognormal_to_normal()`, `normal_to_lognormal()`
Examples

```r
generate_cost_function(stats::qlnorm)
```

---

**generate_weights**  
Generate a weighting table for SMEs based upon their calibration answers

---

**Description**

Generate a weighting table for SMEs based upon their calibration answers

**Usage**

```r
generate_weights(questions, responses)
```

**Arguments**

- `questions` *tidyrisk_question_set* object.
- `responses` *tidyrisk_response_set* object

**Value**

A dataframe of SMEs and their numerical weighting.

**Examples**

```r
NULL
```

---

**get_smes_domains**  
Calculate the prioritized list of domains for a given subject matter expert (SME)

---

**Description**

Given a *tidyrisk_question_set* object and the name and the name of a specific SME of interest, create a vector of the domains in order of priority.

**Usage**

```r
get_smes_domains(sme, questions)
```

**Arguments**

- `sme` Name of the subject matter expert.
- `questions` A *tidyrisk_question_set* object.
is_tidyrisk_question_set

Value

An ordered vector of the domains for the requested SME.

Examples

```r
## Not run:
questions <- read_questions()
get_sme_domains("Sally Expert", questions)
## End(Not run)
```

is_tidyrisk_question_set

Test if the object is a tidyrisk_question_set

Description

This function returns TRUE for tidyrisk_question_set or sub-classes thereof, and FALSE for all other objects.

Usage

`is_tidyrisk_question_set(x)`

Arguments

- `x` An object

Examples

```r
## Not run:
is_tidyrisk_question_set(x)
## End(Not run)
```

is_tidyrisk_response_set

Test if the object is a tidyrisk_response_set

Description

This function returns TRUE for tidyrisk_response_set or sub-classes thereof, and FALSE for all other objects.

Usage

`is_tidyrisk_response_set(x)`
Arguments

x An object

Examples

```r
## Not run:
is_tidyrisk_response_set(x)
## End(Not run)
```

---

lognormal_to_normal Convert lognormal parameters to normal parameters

Description

Given a set of parameters describing a lognormal distribution, return the parameters of the underlying normal distribution.

Usage

lognormal_to_normal(meanlog, sdlog)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meanlog</td>
<td>Mean log.</td>
</tr>
<tr>
<td>sdlog</td>
<td>Standard deviation log.</td>
</tr>
</tbody>
</table>

Value

A list.

See Also

Other distribution fitting functions: `combine_lognorm_trunc()`, `combine_lognorm()`, `combine_norm()`, `fit_capabilities_geomean()`, `fit_capabilities()`, `fit_lognorm_trunc()`, `fit_lognorm()`, `fit_norm_trunc()`, `fit_pois()`, `fit_scenarios_geomean()`, `fit_scenarios()`, `fit_threat_communities()`, `generate_cost_function()`, `normal_to_lognormal()`

Examples

```r
lognormal_to_normal(meanlog=1, sdlog=3)
```
**make_handouts**

Create a set of interview handouts for a SME

**Description**

Creates two MS Word documents. One is an answers document that contains the answers to the calibration questions, the other (with the name of the SME) does not contain answers and is intended to be a visual reference (and possible take away) for the SME.

**Usage**

```r
make_handouts(sme, questions, output_dir, calibration_questions = 10)
```

**Arguments**

- `sme` Name of the SME.
- `questions` `tidyrisk_question_set` object
- `output_dir` Directory to place output.
- `calibration_questions` Number of calibration questions to ask.

**Examples**

```r
## Not run:
questions <- read_questions()
make_handouts("Sally Expert", questions, output_dir = tempdir())
## End(Not run)
```

**make_scorecard**

Create a scorecard for marking progress through domains in an interview

**Description**

Creates a two page PDF with one grid for scenarios and one for capabilities. Each grid contains a square for each domain. An analyst can mark/stamp each domain as it is covered in an interview, gamifying progress.

**Usage**

```r
make_scorecard(sme, questions, output_dir)
make_bingo(sme, questions, output_dir = getwd())
```
Arguments

- **sme**
  Name of SME.
- **questions**
  `tidyrisk_question_set` object.
- **output_dir**
  Directory to place scorecards.

Details

The domains are ordered according to the SME’s expertise profile, ensuring they match the interview order flow.

Value

An invisible null.

Examples

```r
## Not run:
questions <- read_questions()
make_scorecard("Sally Expert", questions, output_dir = tempdir())
## End(Not run)
```

Description

Creates an in-browser slideshow as a visual aid when conducting an interview with a subject matter expert (SME). The slideshow is customized for the SME by placing the domains in the order of preference for that SME.

Usage

```r
make_slides(
  sme,
  questions,
  output_dir,
  assessment_title = "Strategic Risk Assessment"
)
```

Arguments

- **sme**
  Name of the SME being interviewed.
- **questions**
  A `tidyrisk_question_set` object.
- **output_dir**
  Directory location for knitted slides.
- **assessment_title**
  Title of the assessment being performed.
mc_calibration_answers

Value

Invisibly returns the full path to the slide file.

Examples

```r
## Not run:
make_slides("Sally Expert", questions, output_dir = tempdir())

## End(Not run)
```

---

**mc_calibration_answers**

*MetroCare Hospital Calibration Answers*

Description

A dataset of SME answers to calibration questions.

Usage

```r
mc_calibration_answers
```

Format

A data frame with 50 rows and 5 variables:

- **sme**: name of the subject matter expert
- **calibration_id**: unique identifier of the calibration question
- **low**: SME’s low end estimate
- **high**: SME’s high end estimate
- **date**: date of answer

Source

This is hypothetical information. Any similarity to any other entity is completely coincidental.
**mc_capabilities**  
*MetroCare Hospital Capabilities*

**Description**
A dataset of program capabilities.

**Usage**
`mc_capabilities`

**Format**
A data frame with 60 rows and 3 variables:

- `capability_id` unique identifier of the capability
- `domain_id` domain associated with the capability
- `capability` text description of the capability

**Source**
This is hypothetical information. Any similarity to any other entity is completely coincidental.

---

**mc_capability_answers**  
*MetroCare Hospital Capability Answers*

**Description**
A dataset of SME answers to capabilities.

**Usage**
`mc_capability_answers`

**Format**
A data frame with 1 rows and 7 variables:

- `sme` name of the SME
- `capability_id` identifier of the capability
- `low` capability estimate, low
- `high` capability estimate, high
- `date` date of the answer

**Source**
This is hypothetical information. Any similarity to any other entity is completely coincidental.
**mc_capability_parameters_fitted**

*MetroCare Hospital Capability Parameters (fitted)*

---

**Description**

A dataset of sample fitted capability parameters.

**Usage**

`mc_capability_parameters_fitted`

**Format**

A data frame with 300 rows and 10 variables:

- `sme` name of the sme providing the response
- `capability_id` unique identifier
- `date` text description of the threat community
- `capability_func` capability sampling function
- `capability_mean` capability mean
- `capability_sd` capability standard deviation
- `capability_min` capability minimum
- `capability_max` capability maximum
- `low` threat communities capability, high end
- `high` threat communities capability, high end

**Source**

This is hypothetical information. Any similarity to any other entity is completely coincidental.

---

**mc_domains**

*MetroCare Hospital Domains*

---

**Description**

A dataset of program domains.

**Usage**

`mc_domains`
mc_scenarios

Format
A data frame with 15 rows and 4 variables:

- **domain**  domain title
- **description**  descriptive text describing the content of the domain
- **active**  logical flag indicating whether or not the domain is in use
- **domain_id**  unique domain id

Source
This is hypothetical information. Any similarity to any other entity is completely coincidental.

---

mc_scenarios  MetroCare Risk Scenarios

Description
A dataset of sample risk scenarios.

Usage
mc_scenarios

Format
A data frame with 56 rows and 5 variables:

- **scenario_id**  unique identifier
- **scenario**  scenario description
- **threat_id**  threat community id
- **domain_id**  domain id
- **controls**  comma separated list of control ids

Source
This is hypothetical information. Any similarity to any other entity is completely coincidental.
**mc_scenario_answers**  
*MetroCare Hospital Scenario Answers*

**Description**
A dataset of SME answers to scenarios.

**Usage**
mc_scenario_answers

**Format**
A data frame with 1 rows and 7 variables:

- **sme**  name of the SME  
- **scenario_id**  identifier of the scenario  
- **freq_low**  frequency estimate, low  
- **freq_high**  frequency estimate, high  
- **imp_low**  impact estimate, low  
- **imp_high**  impact estimate, high  
- **date**  date of the answer

**Source**
This is hypothetical information. Any similarity to any other entity is completely coincidental.

**mc_scenario_parameters_fitted**  
*MetroCare Hospital Scenario Parameters (fitted)*

**Description**
A dataset of sample fitted scenario parameters.

**Usage**
mc_scenario_parameters_fitted
Format

A data frame with 280 rows and 17 variables:

- **sme**: name of the SME providing the response
- **scenario_id**: unique identifier
- **date**: date of the response
- **impact_func**: function to use for impact sampling
- **impact_meanlog**: threat communities capability, high end
- **impact_sdlog**: type of the threat community
- **impact_min**: action type of the threat community
- **impact_max**: action type of the threat community
- **imp_low**: action type of the threat community
- **imp_high**: action type of the threat community
- **frequency_func**: function to use for frequency sampling
- **frequency_meanlog**: frequency meanlog
- **frequency_sdlog**: frequency standard deviation log
- **frequency_min**: frequency minimum
- **frequency_max**: frequency maximum
- **freq_low**: action type of the threat community
- **freq_high**: action type of the threat community

Source

This is hypothetical information. Any similarity to any other entity is completely coincidental.

---

**mc_sme_top_domains**  
*MetroCare Hospital SME Top Domains*

Description

A dataset of focus domains per SME.

Usage

`mc_sme_top_domains`

Format

A data frame with 35 rows and 3 variables:

- **sme**: SME name
- **key**: index of domain
- **value**: name of domain
**mc_threat_communities**

**Source**

This is hypothetical information. Any similarity to any other entity is completely coincidental.

---

**mc_threat_communities**  *MetroCare Hospital Threat Communities*

**Description**

A dataset of sample threat communities.

**Usage**

`mc_threat_communities`

**Format**

A data frame with 6 rows and 7 variables:

- **threat_community**  text title of the threat community
- **threat_id**  unique identifier
- **definition**  text description of the threat community
- **low**  threat communities capability, low end
- **high**  threat communities capability, high end
- **category**  type of the threat community
- **action_type**  action type of the threat community

**Source**

This is hypothetical information. Any similarity to any other entity is completely coincidental.

---

**mc_threat_parameters_fitted**

*MetroCare Hospital Threat Parameters (fitted)*

**Description**

A dataset of sample fitted threat parameters.

**Usage**

`mc_threat_parameters_fitted`
Format

A data frame with 8 rows and 12 variables:

- **action_type**: action type
- **category**: category
- **definition**: text description of the threat community
- **high**: action type of the threat community
- **low**: type of the threat community
- **threat_community**: text title of the threat community
- **threat_func**: sampling function
- **threat_id**: unique identifier
- **threat_max**: threat maximum capability
- **threat_mean**: threat mean capability
- **threat_sd**: threat capability standard deviation
- **threat_min**: threat capability minimum

Source

This is hypothetical information. Any similarity to any other entity is completely coincidental.

---

**normal_to_lognormal**  
_Convert normal parameters to lognormal parameters_

Description

Given parameters that describe a normal distribution, convert them back to parameters for a lognormal distribution.

Usage

`normal_to_lognormal(normmean, normsd)`

Arguments

- **normmean**: Mean.
- **normsd**: Standard deviation.

Value

A list.
prepare_data

See Also

Other distribution fitting functions: combine_lognorm_trunc(), combine_lognorm(), combine_norm(), fit_capabilities_geomean(), fit_capabilities(), fit_lognorm_trunc(), fit_lognorm(), fit_norm_trunc(), fit_pois(), fit_scenarios_geomean(), fit_scenarios(), fit Threat_communities(), generate_cost_function(), lognormal_to_normal()

Examples

normal_to_lognormal(normmean = 20, normsd = 3)

prepare_data Create one or more quantitative scenarios objects suitable for simulation by 'evaluator'

Description

Given parameters for the scenarios, threat communities, capabilities, and the question set, generate a list of tidyrisk_scenario objects that may be fed into evaluator::run_simulation for Monte Carlo simulation.

Usage

prepare_data(
    scenario_parameters,
    capability_parameters,
    threat_parameters,
    questions
)

Arguments

scenario_parameters
    Scenarios with final parameters defined.

capability_parameters
    Capabilities with final parameters defined.

threat_parameters
    Threat communities with final parameters defined.

questions
    A tidyrisk_question_set object.

Value

A list of one or more tidyrisk_scenario objects.
Examples

```r
suppressPackageStartupMessages(library(dplyr))
data(mc_domains, mc_capabilities, mc_scenarios, mc_sme_top_domains,
calibration_questions, mc_threat_communities)
question_set <- tidyrisk_question_set(mc_domains, mc_scenarios, mc_capabilities,
calibration_questions, mc_sme_top_domains,
mc_threat_communities)
response_set <- tidyrisk_response_set(mc_calibration_answers,
mc_scenario_answers, mc_capability_answers)
sme_weightings <- generate_weights(question_set, response_set)
data(mc_scenario_parameters_fitted, mc_capability_parameters_fitted,
mc_threat_parameters_fitted)
scenario_parameters <- left_join(mc_scenario_parameters_fitted, sme_weightings, by = "sme") %>%
  combine_scenario_parameters()
capability_parameters <- left_join(mc_capability_parameters_fitted, sme_weightings, by = "sme") %>%
  combine_capability_parameters()
quantitative_scenarios <- prepare_data(scenario_parameters,
capability_parameters,
mc_threat_parameters_fitted,
question_set)
```

read_questions

---

**Description**

Reads in all the questions for which subject matter expert input is needed. Includes the domains, capabilities, scenarios, calibration questions, and threat communities.

**Usage**

```r
read_questions(source_dir, active_only = TRUE)
```

**Arguments**

- `source_dir` Directory location to find input files.
- `active_only` Read in only the active elements, defaults to TRUE.

**Details**

Expects the following files to be present:

- `domains.csv` - Domains
  - domain_id, domain
- `capabilities.csv` - Capabilities
  - domain_id, capability_id, capability
- `scenarios.csv` - Scenarios
### Description

Reads in all the responses recorded to the calibration, scenarios, and capability questions.

#### Usage

```r
read_responses(source_dir = getwd())
```

#### Arguments

- `source_dir` Directory location where input files are found.

#### Details

Expects the following files to be present:

- `calibration_answers.csv` - Calibration
- `scenario_answers.csv` - Scenarios
- `capability_answers.csv` - Capabilities

#### Value

A tidyrisk_response_set object
tidyrisk_question_set

Construct a tidyrisk_question_set object

Description

new.tidyrisk_question_set is a low-level constructor that takes a list of dataframes. tidyrisk_question_set constructs a tidyrisk_question_set object from dataframes. as.tidyrisk_question_set is a S3 generic that converts existing objects. validate_tidyrisk_question_set verifies that the data elements are internally consistent.

Usage

tidyrisk_question_set(
    domains,
    scenarios,
    capabilities,
    calibration,
    expertise,
    threat_communities
)

new_tidyrisk_question_set(x)

as.tidyrisk_question_set(x, ...)

validate_tidyrisk_question_set(x)

Arguments

domains    Domains
scenarios   Scenario questions
capabilities Capability questions
calibration  Calibration questions
expertise   SME expertise
threat_communities  Threat communities
x          object to coerce
...        Individual dataframes
tidyrisk_response_set

Examples

NULL

tidyrisk_response_set  Construct a tidyrisk_response_set object

Description

new.tidyrisk_response_set is a low-level constructor that takes a list of dataframes. tidyrisk_response_set constructs a tidyrisk_response_set from dataframes. as.tidyrisk_response_set is a S3 generic that converts existing objects.

Usage

tidyrisk_response_set(
  calibration_answers,
  scenario_answers,
  capability_answers
)

new_tidyrisk_response_set(
  calibration_answers,
  scenario_answers,
  capability_answers
)

as.tidyrisk_response_set(x, ...)

Arguments

calibration_answers  Calibration tidyrisk_response_set
scenario_answers  Scenarios tidyrisk_response_set
capability_answers  Capability tidyrisk_response_set
x  object to coerce
...  Individual dataframes

Examples

NULL
Index

*Topic **datasets**

- calibration_questions, 3
- mc_calibration_answers, 23
- mc_capabilities, 24
- mc_capability_answers, 24
- mc_capability_parameters_fitted, 25
- mc_domains, 25
- mc_scenario_answers, 27
- mc_scenario_parameters_fitted, 27
- mc_scenarios, 26
- mc_sme_top_domains, 28
- mc_threat_communities, 29
- mc_threat_parameters_fitted, 29

as.tidyrisk_question_set
  (tidyrisk_question_set), 34
as.tidyrisk_response_set
  (tidyrisk_response_set), 35

calibration_questions, 3
check_readability, 3
clean_answers, 4
collector, 5
combine_capability_parameters, 5
combine_lognorm, 6, 7, 10–17, 20, 31
combine_lognorm_trunc, 6, 6, 7, 10–17, 20, 31
combine_norm, 6, 7, 10–17, 20, 31
combine_scenario_parameters, 8
derive_controls, 8, 8, 9
enforce_tidyrisk_question_set, 9
enforce_tidyrisk_response_set, 10

fit_capabilities, 6, 7, 10, 11–17, 20, 31
fit_capabilities_geomean, 6, 7, 10, 11, 12–17, 20, 31
fit_lognorm, 6, 7, 10, 11, 12, 13–17, 20, 31

fit_lognorm_trunc, 6, 7, 10–12, 12, 14–17, 20, 31
fit_norm_trunc, 6, 7, 10–13, 13, 14–17, 20, 31
fit_pois, 6, 7, 10–14, 14, 15–17, 20, 31
fit_scenarios, 6, 7, 10–14, 15, 16, 17, 20, 31
fit_scenarios_geomean, 6, 7, 10–15, 16, 17, 20, 31
fit_threat_communities, 6, 7, 10–16, 16, 17, 20, 31
generate_cost_function, 6, 7, 10–17, 17, 20, 31
generate_weights, 18
get_smes_domains, 18

is_tidyrisk_question_set, 19
is_tidyrisk_response_set, 19
lognormal_to_normal, 6, 7, 10–17, 20, 31

make_bingo (make_scorecard), 21
make_handouts, 21
make_scorecard, 21
make_slides, 22
mc_calibration_answers, 23
mc_capabilities, 24
mc_capability_answers, 24
mc_capability_parameters_fitted, 25
mc_domains, 25
mc_scenario_answers, 27
mc_scenario_parameters_fitted, 27
mc_scenarios, 26
mc_sme_top_domains, 28
mc_threat_communities, 29
mc_threat_parameters_fitted, 29

new_tidyrisk_question_set
  (tidyrisk_question_set), 34
new_tidyrisk_response_set
  (tidyrisk_response_set), 35
normal_to_lognormal, 6, 7, 10–17, 20, 30
prepare_data, 31
read_questions, 32
read_responses, 33
run_simulation, 31
tidyrisk_question_set, 9, 18, 21, 22, 31, 33, 34
tidyrisk_response_set, 10, 15, 18, 35
tidyrisk_scenario, 31
validate_tidyrisk_question_set
(tidyrisk_question_set), 34