



CORRUPTION RISK COMPOSITE INDICATOR (CI)

Normalisation, weighting and aggregation procedures

Deliverable WP 3.3

Michela Gnaldi

Niccolò Salvini

Simone Del Sarto

Grant Agreement number: 101038790 — CO.R.E — ISFP-2020-AG-CORRUPT

The content of this document represents the views of the author only and is his/her sole responsibility.

The European Commission does not accept any responsibility for use that may be made of the information it contains.

INDEX OF CONTENTS

Summary.....	3
1. Walkthrough: how to read the ‘coresoi’ R Package.....	4
1.1 Premises	4
1.2 Package Documentation	4
1.3 coresoi.....	5
1.4 Get Started	6
1.5 Reference	7
1.6 Articles (documentation)	8
1.7 Changelog.....	10
2. Normalisation, weighting and aggregation procedures in the <i>coresoi</i> R Package.....	11
2.1 coresoi: computation, documentation and exemplification of the elementary indicators of corruption risk in emergency	12
2.2 coresoi: computation, documentation and exemplification of the normalisation, weighting and aggregation procedures	14



SUMMARY

This document contains the CO.R.E. deliverable 3.3. It is composed of two main sections:

1. Walkthrough: how to read the *coresoi* R Package
2. Normalisation, weighting and aggregation procedures in the *coresoi* R Package.

The Walkthrough is an introductory section intended at supporting the users in the process of reading the documentation for the *coresoi* package, available at <https://core-forge.github.io/coresoi/>.

A Package Documentation is typically organised into different sections that provide information about the R package, such as the purpose of the package, the functions and datasets it provides, and how to use the package effectively.

In the case of *coresoi*, the documentation is made of five sections: “coresoi”, “Get started”, “Reference”, “Articles” and “Changelog”. In the Walkthrough section, the internal architecture of the five sections will be detailed to ease users’ package navigation and practical use.

As the Walkthrough section provides preliminary support to the users, it is included in the first part of the deliverable 3.3 and recalled in the deliverables 3.4 and 3.5.

The subsequent section 2 devoted to the ‘Normalisation, weighting and aggregation procedures in the *coresoi* R Package’, includes the specific content of the deliverable 3.3. Specifically, the section includes two sub-sections:

- sub-section 2.1 devoted to the description of the functions in *coresoi* to compute the elementary indicators of corruption risk in emergency, together with the users’ documentation or guide;
- sub-section 2.2 devoted to the ‘Composite Indicator (CI) of corruption risk in emergencies’ and containing the description of the functions in *coresoi* to perform the normalisation, weighting and aggregation procedures, together with the users’ documentation or guide.



1. WALKTHROUGH: HOW TO READ THE 'CORESOI' R PACKAGE

1.1 Premises

R packages are essential tools for data analysts and statisticians who work with the R programming language. These packages consist of a set of organised functions, data, and documentation that are designed to solve specific data analysis problems.

To use an R package effectively, it is essential to understand its documentation. The documentation contains all the relevant information for a package, including the functions it contains, how to use them, and any other important details.

The *coresoi* package is a valuable tool for quantifying risk in emergency scenarios (such as COVID-19 or earthquakes) by exploiting relevant red flags in open contracting. It provides support to any users interested in computing corruption risk in public procurement over emergencies through analytical codes, users' guides and practical examples. *coresoi* contains a toy dataset based on the Italian Anti-Corruption Authority (ANAC) open data – the BDNCP (*Banca Dati Nazionale dei Contratti Pubblici*) – which researchers can use to calculate our elementary indicators of corruption risk (i.e., red flags) under emergencies and the composite indicator (CI) of corruption risk, resulting from the aggregation of the red flags.

The main purpose of R packages like *coresoi* is to provide users with a set of functions, data, and documentation aimed at addressing specific analytical problems. The package's functions and datasets are meant to be directly accessed by users within their R code. The included documentation offers guidance on its usage. Additionally, R packages like *coresoi* are essential as they are guaranteed to be easily installable across different systems. With regular building, installation, and testing, users can rely on the package's stability and conformity to industry standards.

In this document, we will focus on the process of reading the documentation for the *coresoi* package. The documentation for this package is available at <https://core-forge.github.io/coresoi/>, and we will explore it in detail in the following sections to show how to use the package to compute the risk of corruption in public procurement data under different emergency scenarios.

1.2 Package Documentation

The Package Documentation is typically organised into different sections that provide information about the R package, such as the purpose of the package, the functions and datasets it provides, and how to use the package effectively. Generally, the documentation has an introductory section that explains the motivation

behind the package, the scope of the package, and provides an overview of its features. This section is usually easy to read and provides a general introduction to the package.

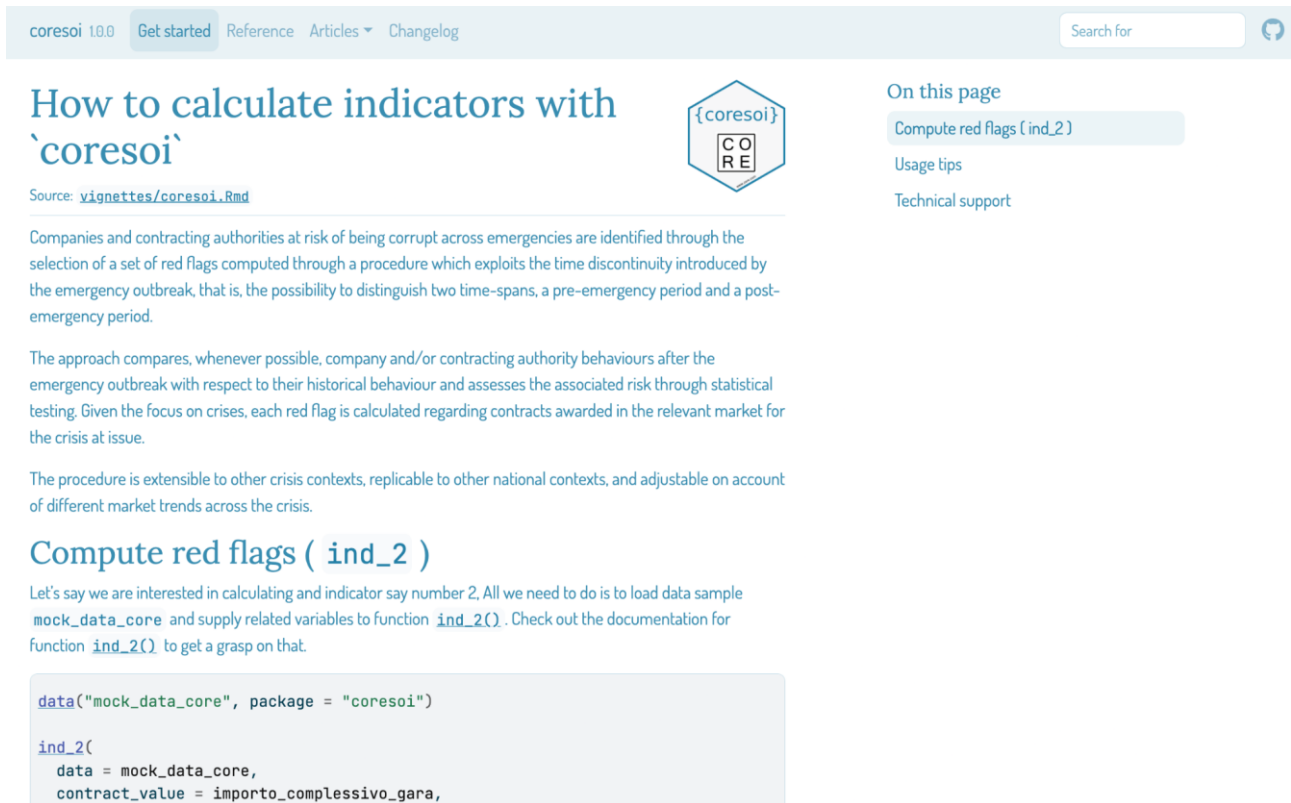
In the case of *coresoi*, the documentation is made of five sections: “coresoi”, “Get Started”, “Reference”, where the functions to compute red flags of corruption risk in emergencies are displayed, “Articles”, where users’ cases and further documentation are located, and “Changelog”.


The internal architecture of the five sections will be detailed in the following, to ease users’ package navigation and practical use.

1.3 coresoi

The “coresoi” section is an introductory section providing synthetic information of both the *coresoi* package and the CO.R.E. project structure, rationale and objectives.

1.4 Get Started



coresoi 1.0.0 Get started Reference Articles ▾ Changelog 

How to calculate indicators with `coresoi`

Source: [vignettes/coresoi.Rmd](#)

Companies and contracting authorities at risk of being corrupt across emergencies are identified through the selection of a set of red flags computed through a procedure which exploits the time discontinuity introduced by the emergency outbreak, that is, the possibility to distinguish two time-spans, a pre-emergency period and a post-emergency period.

The approach compares, whenever possible, company and/or contracting authority behaviours after the emergency outbreak with respect to their historical behaviour and assesses the associated risk through statistical testing. Given the focus on crises, each red flag is calculated regarding contracts awarded in the relevant market for the crisis at issue.

The procedure is extensible to other crisis contexts, replicable to other national contexts, and adjustable on account of different market trends across the crisis.

Compute red flags (ind_2)

Let's say we are interested in calculating an indicator say number 2. All we need to do is to load data sample `mock_data_core` and supply related variables to function `ind_2()`. Check out the documentation for function `ind_2()` to get a grasp on that.

```
data("mock_data_core", package = "coresoi")

ind_2(
  data = mock_data_core,
  contract_value = importo_complessivo_gara,
```

The “Get started” section of an R package documentation is an important part of the documentation. It provides new users with an introduction to the package and helps them to start interacting with the package quickly.

Here are some key elements of an effective “Getting started” section:

1. How to calculate indicators with ‘coresoi’. It provides a tutorial on how to use the package. The tutorial should be seen as an introductory guide that walks the user through the different types of analysis the package can perform.
2. Compute red flags. It includes example codes on how to use the functions (in this case, that for computing indicator 2).
3. Usage tips. It includes tips on how to install, load and use the package effectively. These tips can help new users to avoid common mistakes and make better use of the package.
4. Technical support. It provides information on how users can get help if they encounter problems with the package. This could include links to a help forum or support email address, or instructions on how to report bugs or issues with the package.

In our case, we display a brief dive into the calculation of an elementary indicator (`ind_2` in this case) in the context of the COVID 19 emergency, whose analysis targets are Italian provinces. Data used for this showcase

is still offered by the package under the name of ``mock_data_core``. Results show the first 20 rows of the resulting dataset for ``ind_2``.

1.5 Reference



The screenshot shows the 'Reference' section of the *coresoil* 1.0.0 package documentation. The page title is 'Function reference' and the sub-section is 'Red flags of corruption risk in emergencies'. A sidebar on the right lists 'On this page' with links to 'Red flags of corruption risk in emergencies', 'Composite Indicator (CI) of corruption risk in emergencies', 'Data', and 'Others'. The main content lists six elementary indicators:

- `ind_1()`: Compute Winning rate across the crisis indicator
- `ind_2()`: Compute Awarded economic value across the crisis indicator
- `ind_3()`: Compute Economic deviation across the crisis indicator
- `ind_4()`: Compute Contract Length deviation across the crisis indicator
- `ind_5()`: Compute Winner's share of issuer's contract across the crisis indicator
- `ind_6()`: Compute Communication default across the crisis indicator

The “Reference” is the key section of the *coresoil* package, providing a list of all the functions provided by the package (including also sample data and utility functions), along with a description of each function, the arguments it requires and examples on how to use the functions with sample codes and data at hand.

The “Reference” section is structured into two main sub-sections: the ‘Red flags of corruption risk in emergencies’ sub-section and the ‘Composite Indicator (CI) of corruption risk in emergencies’ sub-section. The former is devoted to the computation, documentation and exemplification of the elementary indicators of corruption risk. The latter is devoted to the computation, documentation and exemplification of the synthetic indicator of corruption risk by company, contracting authority, region, province and municipality. Moreover, two further sub-sections are included: “Data” and “Other”. The former gathers information on the sample data we used to showcase and validate (for internal testing purposes) indicator functions. The latter collects together all the internal functions that constitute the backbone of our software.

The section provides a technical reference and serves as the primary resource for using the package. For example, in sub-section ‘Red flags of corruption risk in emergencies’, the function `'ind_1()'`, computes the first elementary indicator of corruption risk in emergency (or red flag), labelled “Winning rate across the crisis”. Similarly, `'ind_2()'` computes the second red flag, labelled “Awarded economic value across the crisis”.



Once a function has been selected, users can access the documentation (or guide). The users' documentation typically provides information about the function, including its meaning and usage, the arguments it accepts, the default values for those arguments, along with additional information on how to use the function effectively and any special caveats or considerations that users should be aware of when using that function.

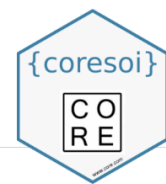
In paragraphs 2.1 and 2.2 of the present document, the functions in *coresoi* to compute the elementary indicators of corruption risk in emergency (together with the users' documentation or guide) and the functions to carry out the normalisation, weighting and aggregation procedures (together with the users' documentation or guide) will be described, respectively.

1.6 Articles (documentation)

summary

Source: [vignettes/articles/summary.f](#)

The objective of this summary is to synthesize the CO.R.E project to the final aim of validating corruption risk in public procurement in emergency.



taken within the
develop a composite indicator of corruption

A major challenge in the measurement of complex and latent phenomena such as corruption and corruption risks consists in summarising information available from a set of single indicators (i.e. red flags) into a single metrics, such as a Composite Indicator (CI) of corruption risk. CIs are a useful communication tool for conveying summary information in a relatively simple way. They are used widely in various sectors in public services as tools in policy analysis and public communication to compare units of analysis (Countries, regions, contracting bodies etc.). Constructing a CI is a difficult task and full of pitfalls: from the obstacles regarding the availability of data and the choice of the individual indicators to their normalization, aggregation, weighting and validity check. Currently, both at the academic and the policy level, little energies have been devoted to developing a robust synthetic/composite indicator of corruption risk in public procurement, as an aggregated measure of single red flags. Among them, the proposal of Fazekas and colleagues (Fazekas et al., 2016) stands out. In this work, the authors develop a composite score of tendering red flags, the Corruption Risk Index (CRI), as a proxy measure of high-level corruption in public procurement, derived from public procurement data from 28 European countries for 2009–2014. A similar study is carried out by Troia (Troia, 2020) with data and red flag indicators developed from the Italian National Dataset of Public Contracts (Banca Dati Nazionale dei Contratti Pubblici). Further, in the Single Market scoreboard Initiative, the single indicators (i.e., red flags) are aggregated by summing them to show how different EU countries are



The “Article” section is where we provide additional information and resources related to the package. This section goes beyond the technical aspects of the package, exploring how the package can be used in various fields, the challenges it addresses, and other topics of interest to the community of users and developers.

The “Article” section usually contains blog posts, papers and other resources that are either authored by the package creators themselves or by the broader community using the package. Here are some examples of the type of content that might be included in the Documentation section:

1. Technical blog posts that provide in-depth technical detail on how to use the package, troubleshoot common issues, or that highlight specific features of the package.
2. Case studies that demonstrate how the package has been used in real-world scenarios.
3. Tutorials and educational materials, workshops, and presentations that teach users about the package or its underlying concepts.
4. Published articles, preprints, or manuscripts that describe research that either uses the package or was used to develop the package.
5. Community contributions from the broader community of developers or users (including blog posts, tutorials, videos or other materials), which help disseminate knowledge about the package and demonstrate how it is being used in various contexts.

In the *coresoi* package, the “Article” section contains an adapted documentation taken from the related CO.R.E documentation (i.e., project deliverables). Specifically, the section includes the following four sub-sections:

- Summary
- Choice of elementary indicators
- Data selection and architecture
- Composite Indicator (CI)
- Try `coresoi` with you own data
- Use CORE API in R

1.7 Changelog



The screenshot shows the Changelog page for coresoi 1.0.0. The page header includes navigation links for coresoi 1.0.0, Get started, Reference, Articles, and Changelog, along with a search bar and a GitHub icon. The main content lists the following changes:

- coresoi 1.0.0**
 - add *composite* main function
 - complete documentation for *composite*
- coresoi 0.2.0**
 - updates pkg documentation
 - release first version for composite indicator
- coresoi 0.1.1**
 - realesed all the feasible red flags: from `ind_1` to `ind_9`
- coresoi 0.1.0**

On the right side, there is a sidebar titled "On this page" with a list of version numbers: 1.0.0, 0.2.0, 0.1.1, and 0.1.0. A hexagonal logo with the text "{coresoi}" and "CORE" is also visible.

The “Changelog” provides a record of changes to the package between different versions, including new features added, bugs fixed, and any other updates made to the package. This helps users who are upgrading to a new version of the package understand what has changed and what new functionality is available. It also helps developers understand the evolution of the package and the motivations behind the changes made over time.

The section includes:

- the version number
- the date of release so that users can keep track of when updates were made
- changes and updates made to the package since the last version
- Impact and migration. For some changes, it may be helpful to include information on how they will impact users and how to migrate to the new functionality. Providing this information can help users better understand the changes and adapt to any modifications.
- Branch-specific changes. Since we managed the whole software development project on an open source online versioning tool (i.e. Github), we are able to trace back any changes applied to our codes. As a result, changes will only be relevant to a particular branch (please if you are unfamiliar with terminology refer to [gitflow](#) and this resource we explicitly develop to collaborate through [Github](#) and [RStudio](#)) of the package (e.g. only certain platforms or only certain packages). In these cases, it should be made clear in the changelog which changes apply to which branches.

2. NORMALISATION, WEIGHTING AND AGGREGATION PROCEDURES IN THE *CORESOI* PACKAGE

coresoi 1.0.0 Get started Reference Articles ▾ Changelog 

Composite Indicator (CI) of corruption risk in emergencies

Synthetic indicator of corruption risk by contracting authority, company, province and municipality (IRT dimensionality assesment)

- `create_indicator_matrix()`
create_indicator_matrix
- `normalise()`
normalise
- `get_weights()`
get_weights
- `aggregate()`
aggregate
- `composite_sensitivity()`
composite_sensitivity
- `dimensionality_check()`
dimensionality_check
- `ind_all()`
ind_all
- `compute_composite()`
compute_composite

On this page

- Red flags of corruption risk in emergencies
- Composite Indicator (CI) of corruption risk in emergencies**
- Data
- Others

Normalisation, weighting and aggregation are necessary steps when developing a composite indicator (CI) of corruption risk. The CI procedure is implemented starting from a baseline of elementary indicators of corruption risk (or red flags). In fact, normalisation has the aim of taking all the red flags to the same scale and to the same polarity. Once the red flags are normalised, they need to be combined to obtain the CI. To this end, weighting and aggregation schemes need to be chosen. The former implies the scale of importance of each individual indicator, while the latter identifies the technique for summarising the individual indicator values into a single number.

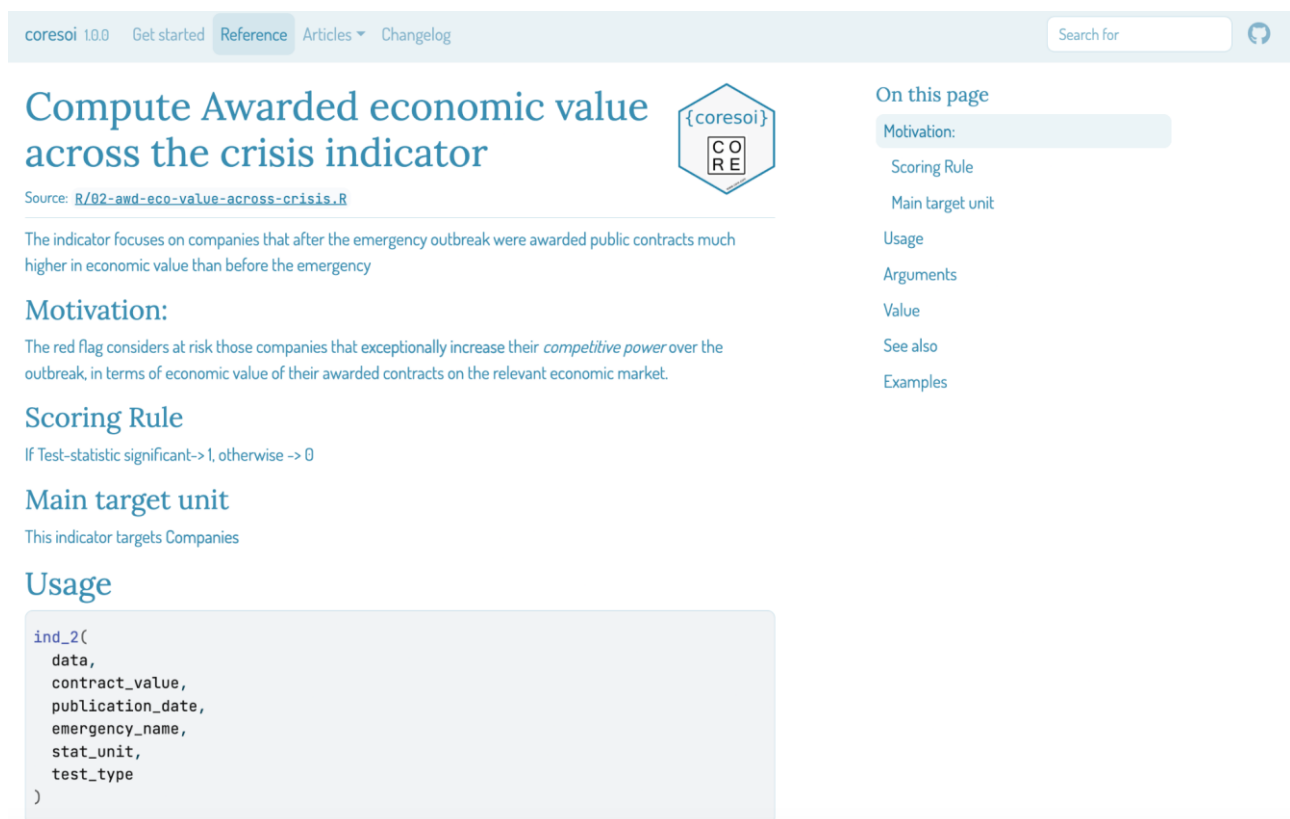
In the following, we will describe the content of the *coresoi* package devoted to i. the computation, documentation and exemplification of the elementary indicators of corruption risk in emergency (red flags), and ii. the computation, documentation and exemplification of normalisation, weighting and aggregation procedures.

As outlined above, the “Reference” section is the most relevant for the *coresoi* package for these tasks. Specifically, the “Red flags of corruption risk in emergencies” sub-section includes the computation,

documentation and exemplification of the elementary indicators of corruption risk in emergency. The “Composite Indicator (CI) of corruption risk in emergencies” sub-section contains the computation, documentation and exemplification of the synthetic indicator of corruption risk by company, contracting authority, region, province and municipality, including the normalisation, weighting and aggregation procedures.

2.1 coresoi: computation, documentation and exemplification of the elementary indicators of corruption risk in emergency

The computation, documentation and exemplification of the elementary indicators of corruption risk in emergency is included in the “Red flags of corruption risk in emergencies” sub-section. Below the interface for `ind_2`.



The screenshot shows the CORE website interface for the indicator 'Compute Awarded economic value across the crisis indicator'. The page includes a navigation bar with 'coresoi 1.0.0', 'Get started', 'Reference', 'Articles', and 'Changelog'. A search bar is present on the right. The main content area features the title, source link, a description, and sections for 'Motivation', 'Scoring Rule', 'Main target unit', and 'Usage'. The 'Usage' section contains a code block for the `ind_2` function. A sidebar on the right lists 'On this page' with links to 'Motivation', 'Scoring Rule', 'Main target unit', 'Usage', 'Arguments', 'Value', 'See also', and 'Examples'.

coresoi 1.0.0 Get started Reference Articles Changelog Search for

Compute Awarded economic value across the crisis indicator

Source: [R/g2-awd-eco-value-across-crisis.R](#)

The indicator focuses on companies that after the emergency outbreak were awarded public contracts much higher in economic value than before the emergency

Motivation:
The red flag considers at risk those companies that exceptionally increase their *competitive power* over the outbreak, in terms of economic value of their awarded contracts on the relevant economic market.

Scoring Rule
If Test-statistic significant-> 1, otherwise -> 0

Main target unit
This indicator targets Companies

Usage

```
ind_2(
  data,
  contract_value,
  publication_date,
  emergency_name,
  stat_unit,
  test_type
)
```

On this page

- Motivation:
- Scoring Rule
- Main target unit
- Usage
- Arguments
- Value
- See also
- Examples

The functions to compute the elementary indicators are the following:

ind_1(): Winning rate across the crisis indicator

ind_2(): Awarded economic value across the crisis indicator

ind_3(): Economic deviation across the crisis indicator

ind_4(): Contract Length deviation across the crisis indicator

ind_5(): Winner's share of issuer's contract across the crisis indicator

ind_6(): Communication default across the crisis indicator

ind_7(): One-shot opportunistic companies over the crisis indicator

ind_8(): Pre-existing contracts modified after the crisis indicator

ind_9(): Lengthy Contracts indicator

Moreover, a wrapper function 'ind_all' is provided with the aim of computing all the elementary indicators at once.

For each function for computing each red flag, the users' documentation/guide includes:

- Description. It provides a quick way for users to understand what each function does and what data are required. In the particular example of the function called ind_1() - which computes the indicator "Winning rate across the crisis" – the description field clarifies what the indicator is intended at measuring. This indicator is designed to 'identify companies that have been awarded public contracts much more frequently after an emergency outbreak than before the emergency'.
- Motivation. It describes the purpose of the function. In the particular example of the function called ind_1(), its purpose is to 'flag companies that have increased their competitive power on the economic market during and after an emergency'.
- Scoring Rule. In the sub-section devoted to the computation of the red flags, it describes how the indicator is scored using a test statistic (i.e., if the test statistic is significant, the score is 1, otherwise, it is 0).
- Main target unit. In the sub-section devoted to the computation of the red flags, it specifies the main unit of measurement for the indicator, which may be companies, contracting authorities, or both.
- Usage. It provides an overview of how to use the function, including the arguments it requires. In the case of the function called ind_1(), the function requires the data, publication_date, emergency_name, stat_unit, and test_type arguments to be specified to calculate the winning rate indicator.
- Arguments. It includes additional details on each of the function's arguments. In the case of the function called ind_1(), the data argument specifies the input data that the function will work on, while the publication_date argument specifies the date of publication of the relevant notice. The emergency_name argument specifies the name of the emergency for which the indicator needs to be computed, while the stat_unit argument specifies the unit of measurement for the indicator. Finally, the test_type argument specifies the test type to be used in the computation of the indicator.

- Value. It describes the output of the function, which is the function schema generated by the `generate_indicator_schema()` function.
- Examples. They provide some sample codes that demonstrate how to practically use the functions, which can be useful for users who are new to the package and are learning how to use it. In the case of `ind_1()` function, the example code shows how to call the function with several arguments and provides an example output if the code blocks were run.

Moreover the reference section may also provide information about datasets provided by the package. In the example provided, there is a dataset called "mock_data_core" which is described as "BDNCP open data from ANAC". Similarly, the function "emergency_dates()" retrieves the emergency date from an input string. This is made even clearer when looking into the reference section, under data sub-menu, for 'mock_data_core'.

2.2 coresoi: computation, documentation and exemplification of the normalisation, weighting and aggregation procedures

The computation, documentation and exemplification of normalisation, weighting and aggregation procedures are included in the 'Composite Indicator (CI) of corruption risk in emergencies' sub-section of the "Reference" section.

The main functions in *coresoi* for performing this task are listed below:

- "normalise", which carries out the normalisation to bring the elementary indicators to the same scale using a suitable normalisation method (e.g., ranking, min-max, dichotomisation);



normalise

Source: [R/composite.R](#)



`normalise` normalises the elementary indicators using a suitable normalisation method (e.g. ranking, min-max, dichotomisation, etc.).

Usage

```
normalise(data, method = "binary", cutoff = 0.95)
```

On this page

Usage

Arguments

Value

Details

Examples

Arguments

data

data matrix of elementary indicators (as returned by [create_indicator_matrix\(\)](#))

method

normalisation method, to be chosen among:

- **"binary"**: each elementary indicator is dichotomised (0/1) using a suitable threshold, to be specified through the argument `cutoff`. Specifically, the normalised indicator will be equal to 1 if the original indicator is greater than the threshold, and 0 otherwise;
- **"ranking"**: each elementary indicator is normalised according to the ranking (see [rank\(\)](#));
- **"z-score"**: each elementary indicator is standardised into z-scores (see [scale\(\)](#)). Let x_{qc} be the original value of elementary indicator q for target unit c . Then, the z-score is obtained as follows:

$$I_{qc} = \frac{x_{qc} - \mu_q(x_{qc})}{\sigma_q(x_{qc})}$$

- **"get_weights"**, which returns the weights for constructing the composite indicator. Three choices are provided: i. equal: each elementary indicator receives the same weight; ii. experts: each elementary indicator receives a specific weight provided by a pool of CO.R.E. experts; iii. irt: each elementary indicator receives specific weights provided by the IRT (Item Response Theory) framework;

get_weights

Source: [R/composite.R](#)



`get_weights` returns the weights for constructing the composite indicator.

Usage

```
get_weights(data, method, ...)
```

On this page

Usage

Arguments

Value

Details

See also

Examples

Arguments

data

data matrix of binary elementary indicators (without missing values).

method

method for getting the set of weights. Possible choices are: "equal", "experts" or "irt". See Details.

...


optional arguments for [mirt::mirt\(\)](#) function. See Details.

Value

vector of weights.




- “aggregate”, which aggregates the set of elementary indicators through the selected method (linear or non-linear) and computes the composite indicator of corruption risk according to the specified set of specified weights;

coresoi 1.0.0 Get started Reference Articles ▾ Changelog 

aggregate

Source: [R/composite.R](#)

aggregate aggregates the set of elementary indicators through the selected method and computes the composite according to the specified set of weights.



Usage

```
aggregate(data, method = "linear", w)
```

Arguments

data
data matrix with the set of normalised elementary indicators (without missing values).

method
aggregation method. Possible choices: "linear" (default) and "non-linear". See Details.

w
vector of weights, as returned by [get_weights\(\)](#).

Value

vector of composite indicator values for each target unit in **data**.

On this page

- Usage
- Arguments
- Value
- Details
- Examples

- “compute_composite”, which directly computes the composite indicator on a set of elementary indicators, by specifying normalisation, weighting and aggregation methods, together with the ways for managing missing values in the indicator.

compute_composite

Source: [R/composite.R](#)

`compute_composite` is a generic function that calculates the composite indicators according to specified *normalisation*, *missing management*, *weighting* and *aggregation* methods.



On this page

Usage

Arguments

Value

Examples

Usage

```
compute_composite(
  indicator_list,
  norm_method = "binary",
  miss_method = 0,
  weight_method = "equal",
  aggr_method = "linear",
  cutoff = 0.95
)
```

Arguments

`indicator_list`

list of outputs about each indicator computable for the target unit (e.g. company or contracting authority), as returned by [ind_all\(\)](#).

`norm_method`

normalisation method (see [normalise\(\)](#)).

`miss_method`

missing management method (see [manage_missing\(\)](#)).

Likewise the functions for computing the elementary indicators, for each of these functions, the users' documentation/guide includes, among others:

- **Description.** It provides a quick way for users to understand what each function does and what data are required.
- **Usage.** It provides an overview of how to use the function, including the arguments it requires.
- **Arguments.** It includes additional details on each of the function's arguments.
- **Value.** It describes the output of the function.
- **Details.** It provides an in-depth and more technical description of what each function does. For example, in the function "get_weights", this part of the documentation specifies the returning objects of the function (i.e., a vector of weights, whose dimension is equal to the number of elementary indicators), the method for obtaining the set of weights (equal, experts and IRT, in this case) and the description of the technical steps to calculate the weights, when necessary (as in the case of weights extracted within the IRT framework).
- **Examples.** They provide some sample codes that demonstrates how to practically use the functions, which can be useful for users who are new to the package and are learning how to use it.