



CORRUPTION RISK COMPOSITE INDICATOR (CI)

Sensitivity

Deliverable WP 3.5

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.....	5
1.3 coresoi.....	5
1.4 Get Started.....	6
1.5 Reference	7
1.6 Articles (documentation).....	9
1.7 Changelog	10
2. Sensitivity analysis of the composite indicator of corruption risk in emergencies in the coresoi R Package	12



SUMMARY

This document contains the CO.R.E. deliverable 3.5. It is composed of two main sections:

1. Walkthrough: how to read the *coresoi* R Package
2. Sensitivity analysis of the composite indicator of corruption risk in emergencies 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 organized 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*, Getting Started, Compute corruption risk in emergencies, Documentation 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 recalled in the first part of the deliverables 3.3, 3.4 and 3.5.

The subsequent section 2 devoted to the "Sensitivity analysis of the composite indicator of corruption risk in emergencies in the *coresoi* R Package" includes the specific content of the deliverable 3.5. Specifically, it includes details on the implementation, documentation and exemplification of the sensitivity analysis.



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.

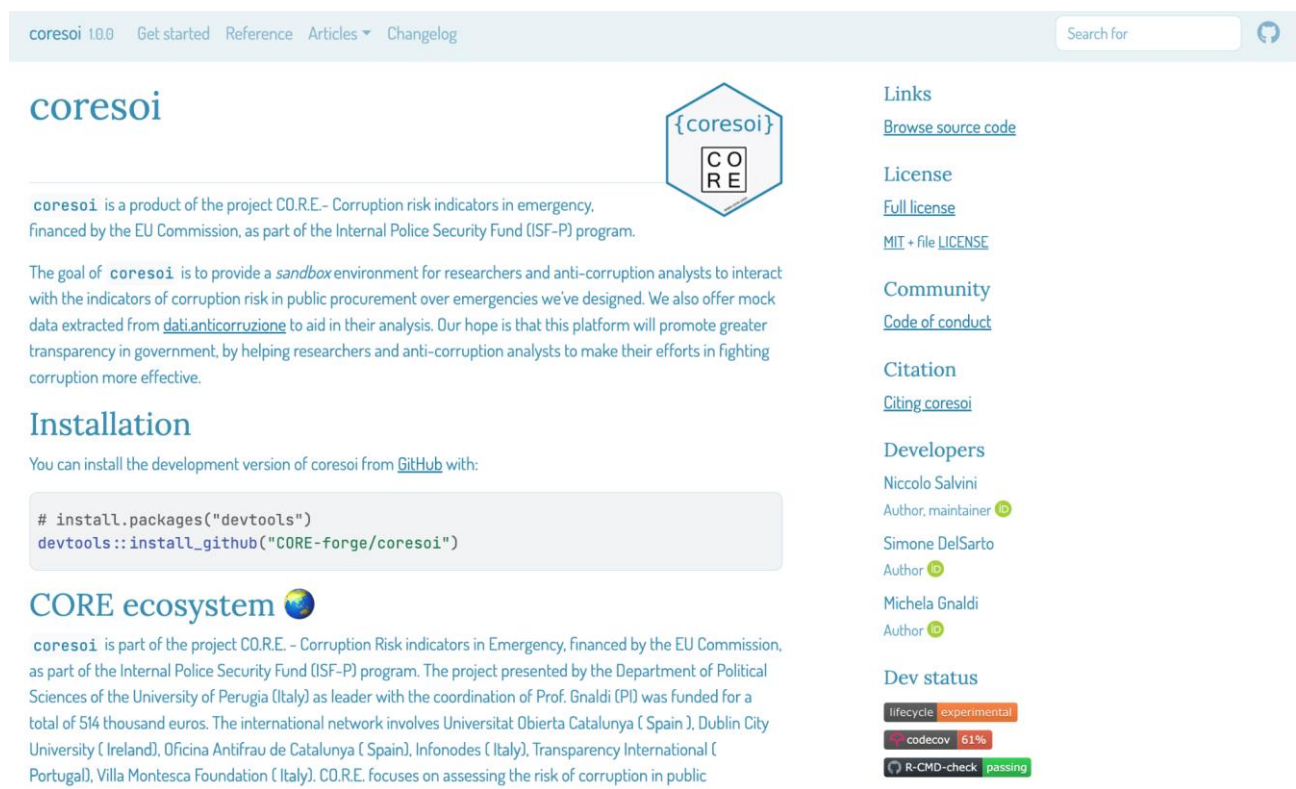
1.2 Package Documentation

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



coresoi 1.0.0 Get started Reference Articles Changelog Search for

coresoi

coresoi is a product of the project CO.R.E.- Corruption risk indicators in emergency, financed by the EU Commission, as part of the Internal Police Security Fund (ISF-P) program.

The goal of **coresoi** is to provide a *sandbox* environment for researchers and anti-corruption analysts to interact with the indicators of corruption risk in public procurement over emergencies we've designed. We also offer mock data extracted from [dati.anticorruzione](#) to aid in their analysis. Our hope is that this platform will promote greater transparency in government, by helping researchers and anti-corruption analysts to make their efforts in fighting corruption more effective.

Installation

You can install the development version of coresoi from [GitHub](#) with:

```
# install.packages("devtools")
devtools::install_github("CORE-forge/coresoi")
```

CORE ecosystem

coresoi is part of the project CO.R.E. - Corruption Risk indicators in Emergency, financed by the EU Commission, as part of the Internal Police Security Fund (ISF-P) program. The project presented by the Department of Political Sciences of the University of Perugia (Italy) as leader with the coordination of Prof. Gnaldi (PI) was funded for a total of 514 thousand euros. The international network involves Universitat Oberta Catalunya (Spain), Dublin City University (Ireland), Oficina Antifrau de Catalunya (Spain), Infonodes (Italy), Transparency International (Portugal), Villa Montesca Foundation (Italy). CO.R.E. focuses on assessing the risk of corruption in public

Links
[Browse source code](#)

License
[Full license](#)
[MIT + file LICENSE](#)

Community
[Code of conduct](#)

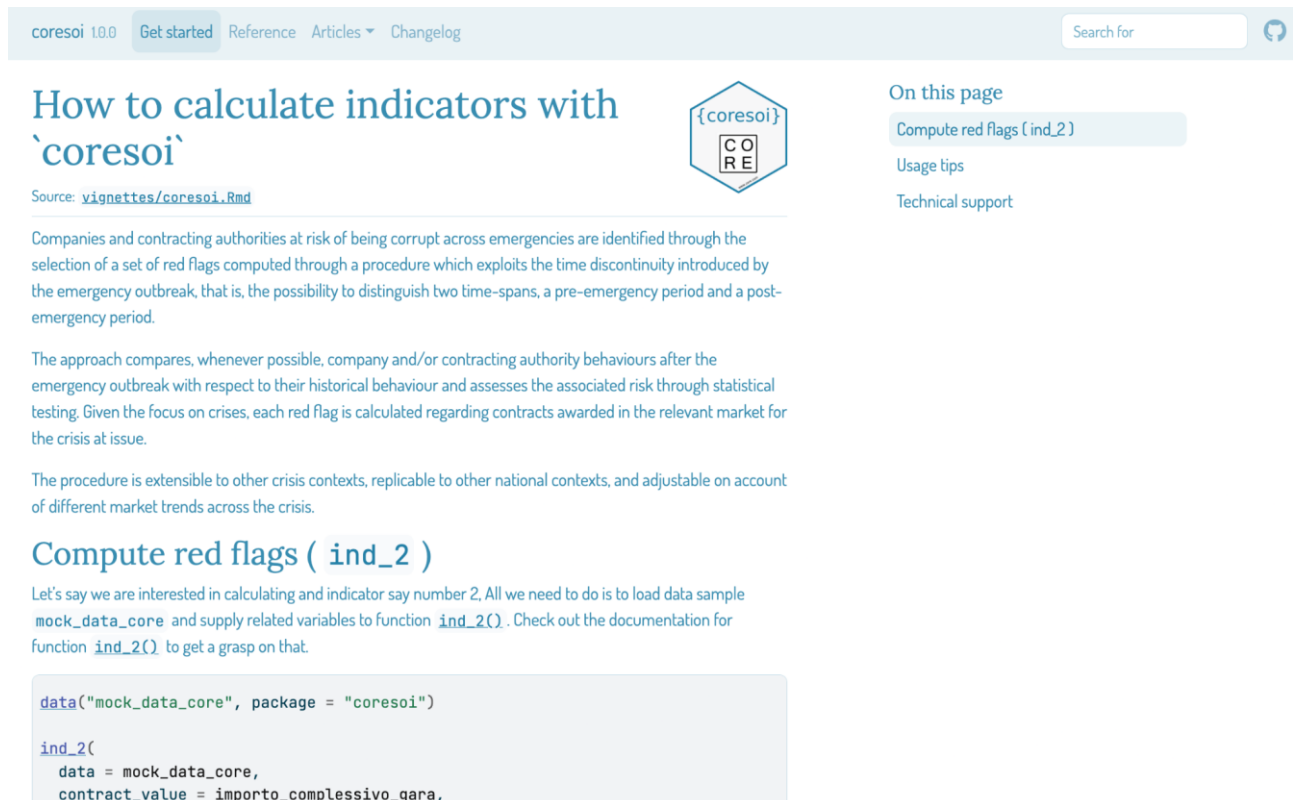
Citation
[Citing coresoi](#)

Developers
Niccolo Salvini
Author, maintainer
Simone DelSarto
Author
Michela Gnaldi
Author

Dev status
lifecycle: experimental
codecov: 61%
R-CMD-check: passing

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



The screenshot shows the documentation page for the 'coresoi' R package. The page title is 'How to calculate indicators with `coresoi`'. The source is cited as 'vignettes/coresoi.Rmd'. The text explains that companies and contracting authorities at risk of being corrupt are identified through a selection of red flags computed through a procedure that exploits time discontinuity. It compares company and/or contracting authority behaviours after an emergency outbreak with their historical behaviour. The procedure is extensible to other crisis contexts. A section titled 'Compute red flags (ind_2)' provides an example of how to load data and use the 'ind_2()' function. The code snippet is as follows:

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

ind_2(
  data = mock_data_core,
  contract_value = importo_compressivo_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:

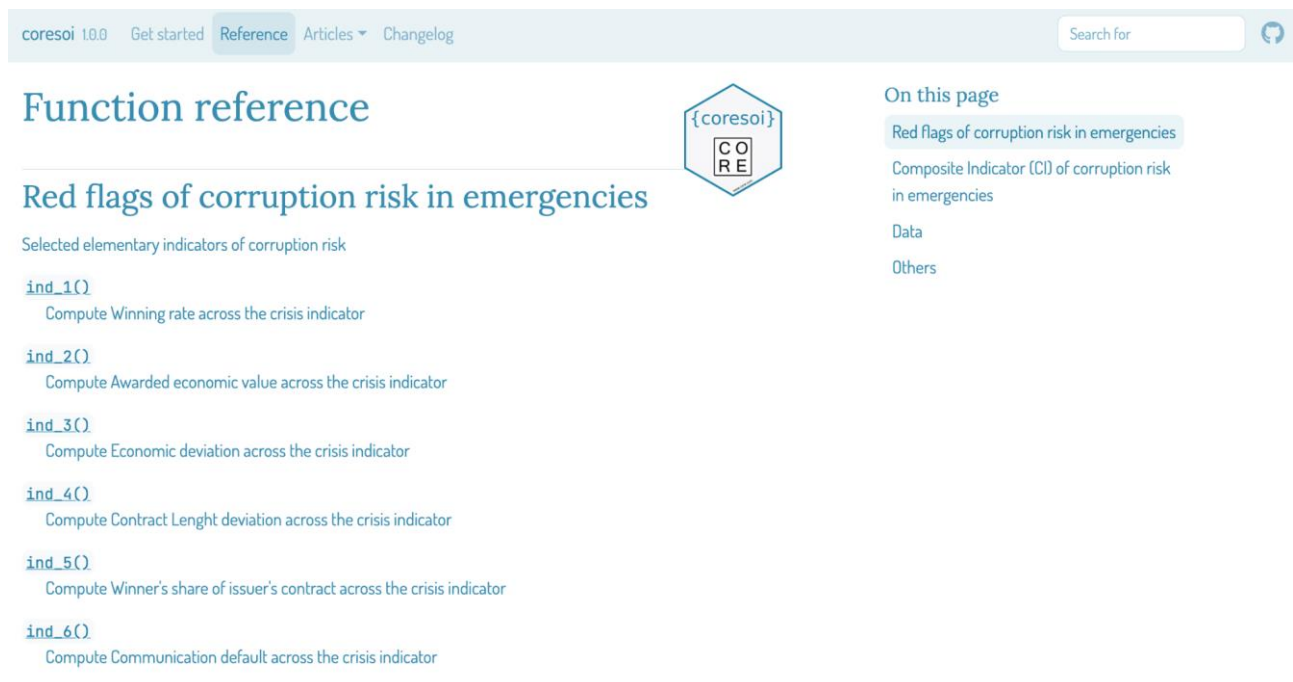
1. How to calculate indicators with ‘coresoi’. It provides a tutorial on how to use the package. The tutorial should be seen as an introductory guide that walks the user through the different types of analysis the package can perform.
2. Compute red flags. It includes example codes on how to use the 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



coresoi 1.0.0 Get started Reference Articles ▾ Changelog 

Function reference

{coresoi}


Red flags of corruption risk in emergencies

Selected elementary indicators of corruption risk

- [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

On this page

- [Red flags of corruption risk in emergencies](#)
- [Composite Indicator \(CI\) of corruption risk in emergencies](#)
- [Data](#)
- [Others](#)

The “Reference” is the key section of the *coresoi* 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.

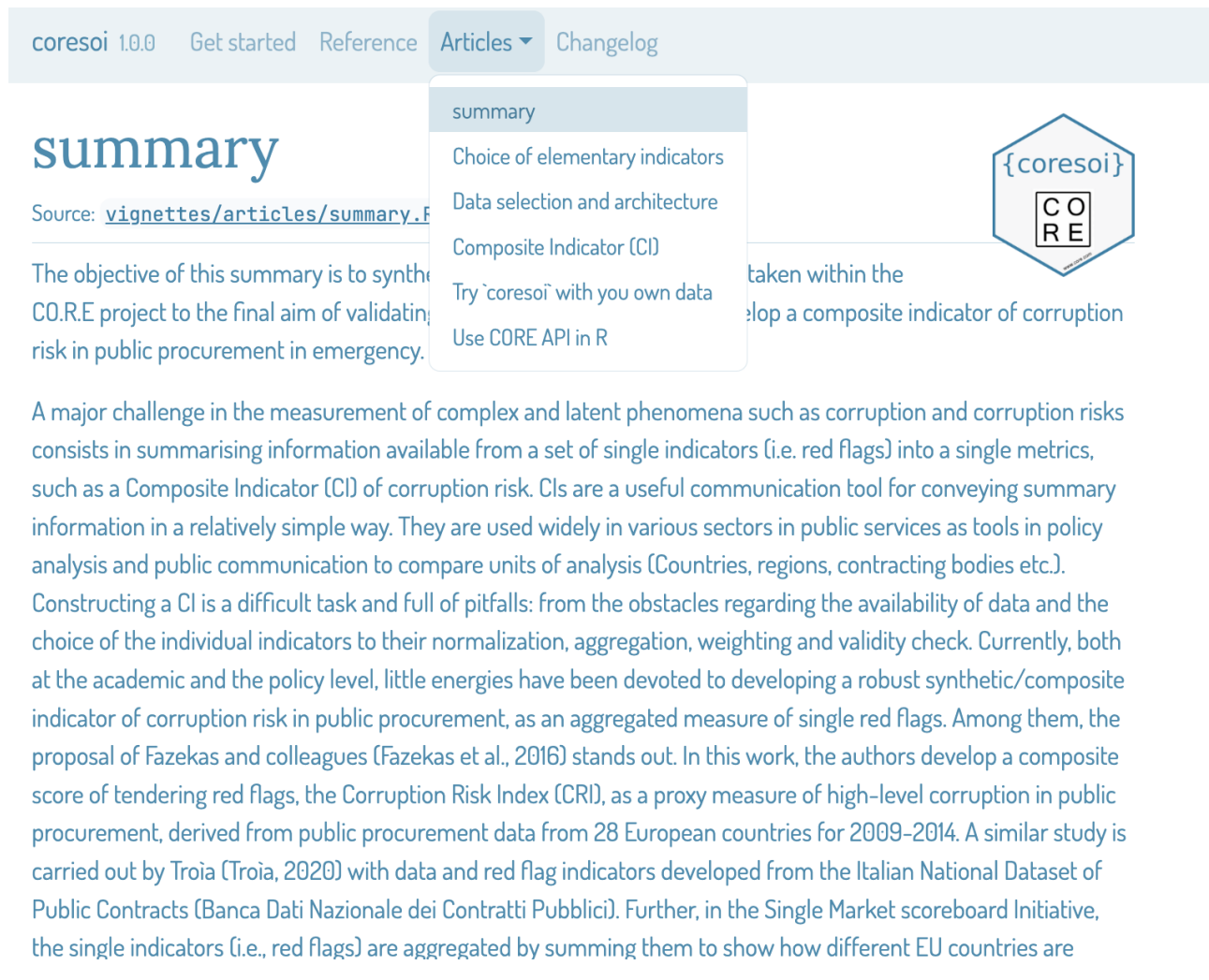
The section provides a technical reference and serves as the primary resource for using the package. For example, in sub-section ‘Red flags of corruption risk in emergencies’, the function ‘ind_1()’, computes the first elementary indicator of corruption risk in emergency (or red flag), labelled “Winning 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 paragraph 2 of the present document, the functions to compute the sensitivity analysis of the composite indicator of corruption risk in emergencies in the *coresoi* R package will be described, together with the users’ documentation or guide.



1.6 Articles (documentation)



The screenshot shows the documentation website for coresoï 1.0.0. The navigation bar includes 'coresoi 1.0.0', 'Get started', 'Reference', 'Articles', and 'Changelog'. The 'Articles' dropdown menu is open, listing: 'summary', 'Choice of elementary indicators', 'Data selection and architecture', 'Composite Indicator (CI)', 'Try `coresoi` with you own data', and 'Use CORE API in R'. The main content area features a 'summary' section with a source link, a paragraph about the objective of the summary, and a detailed paragraph about the challenge of measuring complex phenomena like corruption risk through composite indicators.

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 the *coresoi* 10.0 package. The page has a light blue header with navigation links: *coresoi* 10.0, Get started, Reference, Articles, and Changelog. A search bar is on the right. The main content area features the title "Changelog" and a source link "NEWS.md". A hexagonal logo with "{coresoi}" and "CORE" is displayed. A "On this page" sidebar lists version numbers: 10.0, 0.2.0, 0.1.1, and 0.1.0. The main content lists three versions with their respective 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**

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. SENSITIVITY ANALYSIS OF THE COMPOSITE INDICATOR OF CORRUPTION RISK IN EMERGENCIES IN THE *CORESUI* PACKAGE

coresui 10.0 Get started Reference Articles ▾ Changelog

composite_sensitivity

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

`{coresui}`

`CO`
`RE`

`composite_sensitivity` performs a complete sensitivity analysis of the composite indicator, by computing it using all the possible combinations of methodological choices -- about *normalisation*, *management of missing values* and *weighting* -- as well as evaluating the contribution of each elementary indicator to the final composite, by removing each indicator at a time from the computation. *Note: the unique normalisation method considered within the CORE project is the 'dichotomisation'. Then, the sensitivity is based on the choice of the threshold. See Details.*

Usage

```
composite_sensitivity(indicator_list, cutoff = c(0.9, 0.95, 0.99, 0.995), ...)
```

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()`.

`cutoff`
vector of thresholds for normalising the indicators (i.e., for their dichotomisation)

...
optional arguments of `mirt::mirt()` function (for getting IRT weights).

Value

a list with two versions (wide and long) of a dataframe (`sens_wide` and `sens_long`), which includes, for each target unit, all the possible values of the composite indicator obtained by combining methodological choices and

On this page

- Usage
- Arguments
- Value
- Details
- Examples

The selection of elementary indicators, the choice of normalization, weighting and aggregation schemes, as well as the method for managing missing values in the elementary indicators, are all potential sources of uncertainty that affect both the variance of the composite indicators (CIs) and the variability of any rankings based on them. It is expected that the choice to include a particular indicator in the CI, or the choice to employ a normalization, weighting or aggregation scheme (rather than different ones) can have an impact on the rankings of the individual units according to the CI. In this context, sensitivity analysis is considered as an appropriate tool to assess such uncertainties as it studies how the variation in the output (CI values and/or rankings) can be apportioned to different sources of variation in the assumptions, and how the given CI depends upon the information fed into it. Sensitivity analysis can help to gauge the robustness of the CI value. In particular, the CI can be computed using the combinations of



all the possible alternative choices. For each unit, the distribution of all the possible Cis highlights whether that unit is systematically labeled as corrupt or only under particular circumstances.

In the following, it will be described the content of the *coresoi* package devoted to the implementation, documentation and exemplification of the sensitivity analysis of the composite indicator of corruption risk in emergencies. The ‘Composite indicator (CI) of corruption risk in emergencies’ is the relevant subsection under ‘Function reference’ of the *coresoi* package for this task.

The function that performs the sensitivity analysis of the composite indicator is ‘`composite_sensitivity()`’. It performs a complete sensitivity analysis of the composite indicator, by computing it using all the possible combinations of methodological choices – i.e., normalisation, management of missing values, weighting and aggregation - as well as evaluating the contribution of each elementary indicator to the final composite, by removing each indicator at a time from the computation.

For this function, the users’ documentation/guide includes:

- 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. In the case of the function ‘`composite_sensitivity()`’, it includes: i. `indicator_list`, the list of outputs about each indicator computable for the target unit (e.g., company or contracting authority), as returned by ‘`ind_all()`’; ii. `cutoff`, the vector of thresholds for normalising the indicators (i.e., for their dichotomisation); iii. optional arguments of `mirt::mirt()` function (for getting IRT weights).
- Value. It describes the output of the function. In this case, it returns a list with two versions (wide and long) of a dataframe, which includes, for each target unit, all the possible values of the composite indicator obtained by combining methodological choices and indicator removals.
- Details. It clarifies how the sensitivity procedure is run. Specifically, it clarifies that the main function for carrying out the sensitivity analysis of the composite indicator requires the set of elementary indicators, which need to be included in a list, as returned by the function wrapper ‘`ind_all()`’. This list is given to the internal function ‘`create_indicator_matrix()`’ for obtaining the data matrix of elementary indicators.

Thereafter, several steps for the sensitivity analysis are performed, as follows.

1. Elementary indicators are *normalised* through the ‘dichotomisation’ method, using several thresholds provided through argument ‘`cutoff`’.

2. *Missing values* in the elementary indicators (if any) are managed with both the proposed methods, that is, by replacing missing values with '0' ('not at risk'), or by means of logistic regression models (see internal function `'manage_missing()'`).
3. Vectors of *weights* are obtained according to all the three proposed ways, that is, equal weights, expert weights and IRT weights (see `'get_weights()'`). The last method can require much time, as it depends on the data at hand, which are different every time according to step 1 and 2.
4. *Composite indicator computation*. For each target unit, the composite indicator is computed on the basis of each combination of the above methodological choices (steps 1-3), hence $k \times 2 \times 3$ combinations, where k is the number of normalisation thresholds (cutoff). In addition, the composite indicator is computed by removing each elementary indicator at a time from the computation. Finally, given Q elementary indicators, the composite indicator is computed, for each target unit, $k \times 2 \times 3 \times (Q + 1)$ times.

Results are returned in two dataframes. In the wide version ('sens_wide'), we have the target unit in the rows and as many columns as the number of the above combinations, which report the computed composite. Specifically, the first column is the target unit ID, whereas the subsequent columns contain the composite computed according to the different combinations (steps 1-4 above). The names of these columns have the following structure: `cx.my.w_abc.rrr`, where:

- **x** is the cut-off value for normalising the elementary indicators (e.g., 0.95);
- **y** is the label for missing management method (0 or 1, see `'manage_missing()'`);
- **abc** is the weighting scheme ('eq' for equal weights; 'exp' for expert weights; 'irt' for IRT weights);
- **rrr** is the indication of the removed indicator ('all' means that no indicator is removed).

For example, column labelled as `'c0.95.m0.w_eq.all'` contains the composite indicators computed using: '0.95' as cut-off value for the normalisation; method '0' for missing management; equal weights ('w_eq'); without removal of elementary indicators ('all').

The function directly returns also the long version of the above dataframe, where the target unit is repeated for each 'sensitivity combination'. Here, the columns refer to the variables that enter in the sensitivity analysis. In particular, we have:

- `aggregation_name`: target unit ID
- `ci`: value of the composite
- `cutoff`: possible cut-off values for the normalisation
- `miss`: '0' or '1'
- `weights`: 'eq', 'exp' or 'irt'



- `ind_removed`: 'none', '-ind1', '-ind2', ...,

The long version can be useful for further specific analysis.

- **Examples.** They provide some sample code that demonstrates how to practically use the function, which can be useful for users who are new to the package and are learning how to use it. In the case of the '`composite_sensitivity()`' function, the example code firstly shows the computation of the set of elementary indicators by companies (using '`ind_all()`' wrapper function). This set is given in input to '`composite_sensitivity`', specifying three cutoff values for dichotomising the elementary indicators. A further argument ('TOL') is provided, which makes the computation of the IRT weights more quickly (only for exemplificative purposes). The subsequent code reports an example of possible inspection of the sensitivity analysis output using regression models and ANOVA, in order to evaluate which is the contribution of each phase (normalisation, missing management, weighting and aggregation) to the variability of CI, as well as the contribution of each elementary indicators.