

# Package ‘eudract’

August 9, 2023

**Type** Package

**Title** Creates Safety Results Summary in XML to Upload to EudraCT, or ClinicalTrials.gov

**Date** 2023-07-14

**Version** 0.10.2

**URL** <https://eudract-tool.medschl.cam.ac.uk/>

**BugReports** <https://github.com/shug0131/eudraCT/issues>

**Description** The remit of the European Clinical Trials Data Base (EudraCT <<https://eudract.ema.europa.eu/>>), or ClinicalTrials.gov <<https://clinicaltrials.gov/>>, is to provide open access to summaries of all registered clinical trial results; thus aiming to prevent non-reporting of negative results and provide open-access to results to inform future research. The amount of information required and the format of the results, however, imposes a large extra workload at the end of studies on clinical trial units. In particular, the adverse-event-reporting component requires entering: each unique combination of treatment group and safety event; for every such event above, a further 4 pieces of information (body system, number of occurrences, number of subjects, number exposed) for non-serious events, plus an extra three pieces of data for serious adverse events (numbers of causally related events, deaths, causally related deaths). This package prepares the required statistics needed by EudraCT and formats them into the precise requirements to directly upload an XML file into the web portal, with no further data entry by hand.

**License** GPL-2

**Language** en-GB

**Encoding** UTF-8

**LazyData** true

**Imports** tidy, xslt, dplyr, xml2, utils, magrittr, httr, ggplot2, scales, gridExtra, grid

**Depends** R (>= 3.5.0)

**RoxygenNote** 7.2.3

**Suggests** testthat, knitr, rmarkdown, stringr, covr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Simon Bond [cre],  
Beatrice Pantaleo [aut]

**Maintainer** Simon Bond <simon.bond7@nhs.net>

**Repository** CRAN

**Date/Publication** 2023-08-09 20:50:05 UTC

## R topics documented:

clintrials_gov_convert . . . . .	2
clintrials_gov_upload . . . . .	4
create.safety_summary . . . . .	6
dot_plot . . . . .	6
eudract_convert . . . . .	7
incidence_table . . . . .	9
relative_risk . . . . .	10
safety . . . . .	11
safety_summary . . . . .	12
simple_safety_xml . . . . .	13
soc_code . . . . .	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

clintrials\_gov\_convert

*applies a conversion using xslt from a simple xml file to a ClinicalTrials.gov compatible file, and checks against the schema*

---

### Description

applies a conversion using xslt from a simple xml file to a ClinicalTrials.gov compatible file, and checks against the schema

### Usage

```
clintrials_gov_convert(
  input,
  original,
  output,
  xslt = system.file("extdata", "simpleToCtGov.xslt", package = "eudract"),
  schema_input = system.file("extdata", "simple.xsd", package = "eudract"),
  schema_results = system.file("extdata", "RRSUploadSchema.xsd", package = "eudract"),
  schema_output = system.file("extdata", "ProtocolRecordSchema.xsd", package = "eudract"),
  soc = system.file("extdata", "soc.xml", package = "eudract")
)
```

**Arguments**

input	a character string giving the file path to the simple xml file
original	a character string giving the file path to the study file downloaded from ClinicalTrials.gov
output	a character string naming the output file
xslt	a character string giving the file path to the xslt script. Defaults to the script provided in this package
schema_input	a character string giving the file path to the schema for the simple xml file. Defaults to the schema provided in this package
schema_results	a character string giving the file path to the schema for the results section of the output. A copy was downloaded and is provided in this package as the default.
schema_output	a character string giving the file path to the schema for the overall output. A copy was downloaded and is provided in this package as the default.
soc	a character string giving an xml file that contains the System Organ Class lookup table going from EudraCT numbers to ClinicalTrials words.

**Value**

the output from the validation against the schema. A new file is created as a side-effect, which is suitable to upload into ClinicalTrials.gov. This over-writes the file given in `original` with the additional safety events.

**See Also**

[safety\\_summary\\_simple\\_safety\\_xml](#)

**Examples**

```
safety_statistics <- safety_summary(safety,
                                   exposed=c("Experimental"=60, "Control"=67))
simple <- tempfile(fileext = ".xml")
eudract <- tempfile(fileext = ".xml")
ct <- tempfile(fileext = ".xml")
simple_safety_xml(safety_statistics, simple)
eudract_convert(input=simple,
                output=eudract)
clintrials_gov_convert(input=simple,
                       original=system.file("extdata", "1234.xml", package = "eudract"),
                       output=ct)

## Not run:
# This needs a real user account to work
clintrials_gov_upload(
  input=simple,
  orgname="CTU",
  username="Student",
  password="Guinness",
  studyid="1234"
)
```

```
## End(Not run)
```

---

`clintrials_gov_upload` *applies a conversion using xslt from a simple xml file to a ClinicalTrials.gov compatible file, merges into a study record from the portal, and uploads the result.*

---

### Description

applies a conversion using xslt from a simple xml file to a ClinicalTrials.gov compatible file, merges into a study record from the portal, and uploads the result.

### Usage

```
clintrials_gov_upload(  
  input,  
  orgname,  
  username,  
  password,  
  studyid,  
  url = "https://register.clinicaltrials.gov/",  
  check = interactive(),  
  output = "study_file.xml",  
  backup = "bak_study_file.xml",  
  xslt = system.file("extdata", "simpleToCtGov.xslt", package = "eudract"),  
  schema_input = system.file("extdata", "simple.xsd", package = "eudract"),  
  schema_results = system.file("extdata", "RRSUploadSchema.xsd", package = "eudract"),  
  schema_output = system.file("extdata", "ProtocolRecordSchema.xsd", package = "eudract"),  
  soc = system.file("extdata", "soc.xml", package = "eudract")  
)
```

### Arguments

<code>input</code>	a character string giving the file path to the simple xml file
<code>orgname</code>	a character string giving the organisation name used to log in
<code>username</code>	a character string giving the user-name used to log in
<code>password</code>	a character string giving the password used to log in
<code>studyid</code>	a character string given the unique study id within the portal
<code>url</code>	a character string giving the URL of the website to log in. Defaults to the live site, but the testing site is the alternative.
<code>check</code>	a logical that will check on the command line if you want to proceed as results will be overwritten. Defaults to TRUE.
<code>output</code>	a character string naming the output xml file

backup	a character string naming the copy of the original data that is created.
xslt	a character string giving the file path to the xslt script. Defaults to the script provided in this package
schema_input	a character string giving the file path to the schema for the simple xml file. Defaults to the schema provided in this package
schema_results	a character string giving the file path to the schema for the results section of the output. A copy was downloaded and is provided in this package as the default.
schema_output	a character string giving the file path to the schema for the overall output. A copy was downloaded and is provided in this package as the default.
soc	a character string giving an xml file that contains the System Organ Class look-up table going from EudraCT numbers to ClinicalTrials words.

### Value

Invisibly returns the results from the two API with the portal [response, POST](#) . A new file is created as a side-effect, which is uploaded into ClinicalTrials.gov. This over-writes the original safety data online with the additional safety events. A backup copy of the original data is also saved.

### See Also

[safety\\_summary\\_simple\\_safety\\_xml](#) [ClinicalTrials.gov manual](<https://prsinfo.clinicaltrials.gov/prs-users-guide.html#section10>)

### Examples

```
safety_statistics <- safety_summary(safety,
                                   exposed=c("Experimental"=60,"Control"=67))
simple <- tempfile(fileext = ".xml")
eudract <- tempfile(fileext = ".xml")
ct <- tempfile(fileext = ".xml")
simple_safety_xml(safety_statistics, simple)
eudract_convert(input=simple,
                output=eudract)
clintrials_gov_convert(input=simple,
                       original=system.file("extdata", "1234.xml", package = "eudract"),
                       output=ct)

## Not run:
# This needs a real user account to work
clintrials_gov_upload(
  input=simple,
  orgname="CTU",
  username="Student",
  password="Guinness",
  studyid="1234"
)

## End(Not run)
```

---

`create.safety_summary` *function that creates a safety\_summary object from individual data.frames*

---

### Description

function that creates a `safety_summary` object from individual `data.frames`

### Usage

```
create.safety_summary(group, non_serious, serious)
```

### Arguments

<code>group</code>	a data frame that contains the group-level statistics
<code>non_serious</code>	a data frame that contains the non-serious term-group level statistics
<code>serious</code>	a data frame that contains the serious term-group level statistics

### Value

a `safety_summary` object

---

`dot_plot` *creates a dot-plot of safety data showing the absolute and relative risks*

---

### Description

creates a dot-plot of safety data showing the absolute and relative risks

### Usage

```
dot_plot(  
  safety,  
  type = c("non_serious", "serious"),  
  reference = safety$GROUP$title[1],  
  size = 95,  
  text_width = 10,  
  base = 2,  
  valid_estimates = TRUE  
)
```

**Arguments**

safety	an object created by <a href="#">safety_summary</a> or by <a href="#">relative_risk</a> , in case you want to re-order or filter the choice of rows.
type	a choice of "non_serious" (default) or "serious" as to which type of AE to report on
reference	character vector nameing the referrence arm for teh calculations. Defaults to the first row of the safety\$GROUP.
size	a number between 0-100, giving the size of the confidence interval. Default is 95.
text_width	Integer giving a target width to which the labels are wrapped. Defaults to 10.
base	numeric value to which a log scale uses as tick marks. Suggest powers of 2, or 5.
valid_estimates	a logical, which determines if only terms with valid estimates of relative risk are included in the table. The alternative is to include terms with zeroes.

**Value**

a graphical object that shows the estimates and CI of relative and absolute risk.

**See Also**

[safety\\_summary](#) [relative\\_risk](#) [[relative\\_risks\(\)](#)]

**Examples**

```
safety_statistics <- safety_summary(safety,
  exposed=c("Experimental"=60,"Control"=67))
head( relative_risk(safety_statistics, type="serious") )
dot_plot(safety_statistics, type="non_serious", base=4)
```

---

eudract_convert	<i>applies a conversion using xslt from a simple xml file to a eudract compatible file, and checks against the schema</i>
-----------------	---

---

**Description**

applies a conversion using xslt from a simple xml file to a eudract compatible file, and checks against the schema

**Usage**

```
eudract_convert(
  input,
  output,
  xslt = system.file("extdata", "simpleToEudraCT.xslt", package = "eudract"),
  schema_input = system.file("extdata", "simple.xsd", package = "eudract"),
  schema_output = system.file("extdata", "adverseEvents.xsd", package = "eudract")
)
```

**Arguments**

input	a character string giving the file path to the simple xml file
output	a character string naming the output file
xslt	a character string giving the file path to the xslt script. Defaults to the script provided in this package
schema_input	a character string giving the file path to the schema for the simple xml file. Defaults to the schema provided in this package
schema_output	a character string giving the file path to the schema. A copy was downloaded and is provided in this package as the default.

**Value**

the output from the validation against the schema. A new file is created as a side-effect, which is suitable to upload into eudraCT.

**See Also**

[safety\\_summary](#) [simple\\_safety\\_xml](#)

**Examples**

```
safety_statistics <- safety_summary(safety,
                                   exposed=c("Experimental"=60,"Control"=67))
simple <- tempfile(fileext = ".xml")
eudract <- tempfile(fileext = ".xml")
ct <- tempfile(fileext = ".xml")
simple_safety_xml(safety_statistics, simple)
eudract_convert(input=simple,
                output=eudract)
clintrials_gov_convert(input=simple,
                       original=system.file("extdata", "1234.xml", package = "eudract"),
                       output=ct)

## Not run:
# This needs a real user account to work
clintrials_gov_upload(
  input=simple,
  orgname="CTU",
  username="Student",
  password="Guinness",
```



```

    studyid="1234"
  )

## End(Not run)

```

---

incidence_table	<i>provide standard structured tables to report incidence rates of AEs by arm</i>
-----------------	---

---

### Description

provide standard structured tables to report incidence rates of AEs by arm

### Usage

```
incidence_table(safety, type = c("non_serious", "serious"), percent_round = 0)
```

### Arguments

safety	an object created by <a href="#">safety_summary</a>
type	a choice of "non_serious" (default) or "serious" as to which type of AE to report on
percent_round	integer giving the number of decimal places to round the incidence percentage. Default of 0. Maybe you need more if there is a large sample size and a rare event of interest

### Value

a data.frame that can be directly printed as a table to a report. Each arm has its own column and the text contains "p o is the number of occurrences, and p a percentage of participants with the term.

### See Also

[safety\\_summary](#)

### Examples

```

safety_statistics <- safety_summary(safety,
  exposed=c("Experimental"=60,"Control"=67))
head( incidence_table(safety_statistics, type="serious") )

```

---

relative_risk	<i>Calculate relative risks to be reported or plotted as dot plot</i>
---------------	---

---

### Description

Calculate relative risks to be reported or plotted as dot plot

### Usage

```
relative_risk(
  safety,
  type = c("non_serious", "serious"),
  reference = safety$GROUP$title[1],
  size = 95
)

relative_risk_table(
  safety,
  type = c("non_serious", "serious"),
  reference = safety$GROUP$title[1],
  size = 95,
  digits = 3,
  valid_estimates = TRUE
)

order_filter(rel_risk, threshold = 10)
```

### Arguments

safety	an object created by <a href="#">safety_summary</a>
type	a choice of "non_serious" (default) or "serious" as to which type of AE to report on
reference	character vector nameing the referrence arm for teh calculations. Defaults to the first row of the safety\$GROUP.
size	a number between 0-100, giving the size of the confidence interval. Default is 95.
digits	integer giving the number of significant figures to report to. Default of 3.
valid_estimates	a logical, which determines if only terms with valid estimates of relative risk are included in the table. The alternative is to include terms with zeroes.
rel_risk	a relative risk object
threshold	a threshold on the percent scale, the max percentage for a term the incidence rate needs to exceed

**Value**

a data.frame that has the relative risk estimate and confidence intervals. No adjustment made to deal with zeroes.

A data frame that is suitable for printing to a report, giving relative risks

a revised relative risk object, with the terms concatenated with SOC if there are any duplicates, then ordered by relative risk, into a factor, and filtered to only those terms with an incidence rate above the threshold.

**See Also**

[safety\\_summary dot\\_plot](#)

**Examples**

```
safety_statistics <- safety_summary(safety,
  exposed=c("Experimental"=60,"Control"=67))
head( relative_risk(safety_statistics, type="serious") )
relative_risk_table(safety_statistics, type="serious")
rr <- relative_risk(safety_statistics)
rr2 <- order_filter(rr, threshold=2)
dot_plot(rr2)
```

---

safety

*Example of safety data*

---

**Description**

A dataset containing some example data of safety event in raw source format

**Usage**

safety

**Format**

a data frame with 8 columns and 16 rows

**pt** meddra preferred term code

**subjid** a unique subject identifier

**related** a logical indicating if the event is related to the treatment

**soc** the meddra code for the System Organ Class

**fatal** a numerical 0/1 to indicate if the event was fatal

**serious** a numerical 0/1 to indicate if the event was serious

**group** the treatment group for the subject

**term** a text description of the event. Needs to be matching 1-1 with the pt code

## Details

The data contains one row per patient-event. So the numbers exposed in each arm cannot be inferred from these data, as patients with no events will not be included in these data.

The variable names and formats are those required by [safety\\_summary](#). The variable `pt` is not strictly required. An alternative to `soc` would be the equivalent character string from [soc\\_code](#)

---

safety_summary	<i>Calculate frequency tables from a rectangular data frame with one row per subject-event</i>
----------------	--

---

## Description

Calculate frequency tables from a rectangular data frame with one row per subject-event

## Usage

```
safety_summary(
  data,
  exposed,
  excess_deaths = 0,
  freq_threshold = 0,
  soc_index = c("meddra", "soc_term"),
  na.action = na.fail
)
```

## Arguments

<code>data</code>	a data set containing the following columns: <code>subjid</code> , <code>term</code> , <code>soc</code> , <code>serious</code> , <code>related</code> , <code>fatal</code> , <code>group</code> . See <a href="#">safety</a> for more details.
<code>exposed</code>	a numeric vector giving the numbers of subjects exposed in each group. This needs to be supplied directly by the user, and cannot be inferred from the input data with one row per patient-event. To ensure the ordering is correct either, name the vector with names matching the values in <code>data\$group</code> , or ensure that the <code>data\$group</code> is an ordered factor, or relying on alphabetical ordering of the values in <code>data\$group</code>
<code>excess_deaths</code>	a numeric vector giving the number of extra deaths not reported within data. Defaults to 0.
<code>freq_threshold</code>	a value on a percentage scale at which to remove events if the incidence falls below. Defaults to 0
<code>soc_index</code>	a character vector either <code>"meddra"</code> or <code>"soc_term"</code> , which is used to identify if the <code>soc</code> variable in data gives the numerical meddra code or the description in English.
<code>na.action</code>	a function that indicates what should happen if the data contain missing values. The default is <code>na.fail</code> as both repositories will not accept any missing values in the upload. Alternatives could be <code>na.omit</code> , <code>na.exclude</code> , or <code>na.pass</code> .

**Value**

a list of three dataframes: GROUP, SERIOUS, NON\_SERIOUS. Each contains the summary statistics required by EudraCT, and is suitable for export.

**See Also**

[eudract\\_convert simple\\_safety\\_xml](#)

**Examples**

```
safety_statistics <- safety_summary(safety,
                                   exposed=c("Experimental"=60, "Control"=67))
simple <- tempfile(fileext = ".xml")
eudract <- tempfile(fileext = ".xml")
ct <- tempfile(fileext = ".xml")
simple_safety_xml(safety_statistics, simple)
eudract_convert(input=simple,
                output=eudract)
clintrials_gov_convert(input=simple,
                       original=system.file("extdata", "1234.xml", package = "eudract"),
                       output=ct)

## Not run:
# This needs a real user account to work
clintrials_gov_upload(
  input=simple,
  orgname="CTU",
  username="Student",
  password="Guinness",
  studyid="1234"
)

## End(Not run)
```

---

simple\_safety\_xml      *creates a simple xml file from the input of a safety\_summary object*

---

**Description**

creates a simple xml file from the input of a safety\_summary object

**Usage**

```
simple_safety_xml(
  x,
  file,
  schema = system.file("extdata", "simple.xsd", package = "eudract")
)
```

**Arguments**

x	an object of class <code>safety_summary</code> , as created by <code>safety_summary</code> .
file	a character string name the file to be created
schema	a character string giving the file path to the schema for the outputxml file. Defaults to the schema provided in this package.

**Value**

no output is returned, but a file is created as a side-effect.

**See Also**

[eudract\\_convert\\_safety\\_summary](#)

**Examples**

```
safety_statistics <- safety_summary(safety,
                                   exposed=c("Experimental"=60, "Control"=67))
simple <- tempfile(fileext = ".xml")
eudract <- tempfile(fileext = ".xml")
ct <- tempfile(fileext = ".xml")
simple_safety_xml(safety_statistics, simple)
eudract_convert(input=simple,
                output=eudract)
clintrials_gov_convert(input=simple,
                       original=system.file("extdata", "1234.xml", package = "eudract"),
                       output=ct)

## Not run:
# This needs a real user account to work
clintrials_gov_upload(
  input=simple,
  orgname="CTU",
  username="Student",
  password="Guinness",
  studyid="1234"
)

## End(Not run)
```

---

soc\_code

*System Organ Class coding*

---

**Description**

A dataset containing text descriptions and medDRA and EudraCT codes for each system organ class

*soc\_code*

15

### **Usage**

*soc\_code*

### **Format**

a data frame with 3 columns and 27 rows

**soc\_term** a text description

**euctId** the eudraCT coding

**meddra** the meddra code

### **Source**

<https://www.meddra.org/>, <https://spor.ema.europa.eu/rmswi/#/>

# Index

## \* datasets

safety, [11](#)  
soc\_code, [14](#)

clintrials\_gov\_convert, [2](#)  
clintrials\_gov\_upload, [4](#)  
create\_safety\_summary, [6](#)

dot\_plot, [6](#), [11](#)

eudract\_convert, [7](#), [13](#), [14](#)

incidence\_table, [9](#)

na.exclude, [12](#)

na.fail, [12](#)

na.omit, [12](#)

na.pass, [12](#)

order\_filter(relative\_risk), [10](#)

POST, [5](#)

relative\_risk, [7](#), [10](#)

relative\_risk\_table(relative\_risk), [10](#)

response, [5](#)

safety, [11](#), [12](#)

safety\_summary, [3](#), [5](#), [7-12](#), [12](#), [14](#)

simple\_safety\_xml, [3](#), [5](#), [8](#), [13](#), [13](#)

soc\_code, [12](#), [14](#)