

# Package ‘cancerradarr’

February 16, 2026

**Title** Cancer RADAR Project Tool

**Version** 3.0.0

**Description** Cancer RADAR is a project which aim is to develop an infrastructure that allows quantifying the risk of cancer by migration background across Europe. This package contains a set of functions cancer registries partners should use to reshape 5 year-age group cancer incidence data into a set of summary statistics (see Boyle & Parkin (1991, ISBN:978-92-832-1195-2)) in lines with Cancer RADAR data protections rules.

**License** GPL (>= 3)

**Depends** R (>= 4.1.0)

**Imports** dplyr (>= 1.1.0), epitools, gtools, magrittr, openxlsx (>= 4.2.7), plyr, purrr, rlang, rmarkdown, stats, stringr, tidyr, utils

**Suggests** plotly, shiny, tidyverse, DT, testthat (>= 3.0.0), knitr, quarto

**VignetteBuilder** quarto

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**NeedsCompilation** no

**Author** Nienke Alberts [aut],  
Damien Georges [aut, cre] (ORCID:  
<<https://orcid.org/0000-0003-2425-7591>>),  
Stefano Rosso [aut],  
Iacopo Baussano [aut] (ORCID: <<https://orcid.org/0000-0002-7322-1862>>)

**Maintainer** Damien Georges <[georgesd@iarc.who.int](mailto:georgesd@iarc.who.int)>

**Repository** CRAN

**Date/Publication** 2026-02-16 14:10:10 UTC

## Contents

age_standardized_incidence_rates . . . . .	2
aggregated_ageg_name . . . . .	3
chopped.vector.list . . . . .	4
chop_vector . . . . .	4
create_canradar_summary_file . . . . .	5
create_registry_input_file . . . . .	6
create_static_report . . . . .	7
custom_ageg_aggregation . . . . .	7
dat.aggr . . . . .	9
dat.asr.cat . . . . .	9
dat.cob . . . . .	10
diagnose_input_file . . . . .	10
globocan.2022.eu . . . . .	11
incidence_rates . . . . .	12
incidence_rates_difference . . . . .	13
incidence_rates_ratio . . . . .	14
indirect_proportional_incidence_ratio . . . . .	15
indirect_standardized_incidence_ratio . . . . .	16
open_canradar_dictionary . . . . .	17
proportional_rates . . . . .	17
read_cancerradar_output_01 . . . . .	18
read_cancerradar_output_02 . . . . .	19
read_cancerradar_output_03 . . . . .	20
run_dynamic_report . . . . .	21
standardized_incidence_rate_difference . . . . .	22
standardized_incidence_rate_ratio . . . . .	23
validate_age_groups . . . . .	24
<b>Index</b>	<b>26</b>

---

age\_standardized\_incidence\_rates

*Age-standardized incidence rate (asir)*

---

### Description

Age-standardized incidence rate (asir)

### Usage

```
age_standardized_incidence_rates(ncan, py, pystd, ncan.min = 5, py.min = 0)
```

**Arguments**

ncan	integer, (age-specific) number of cancers in the population of interest
py	integer, (age-specific) person-year in the the population of interest
pystd	numeric, (age-specific) standard population person-years (e.g. standard world population)
ncan.min	integer, minimum number of observation required not to mask the CI's out
py.min	integer, minimum person-years required not to mask the CI's out

Age-standardized incidence rate (asir) and associated 95% confidence interval are computing assuming normal distribution of the asir. asir is a summary statistics that should be computed per group of individuals providing age specific counts.

**Value**

a 1 line and 3 column data.frame containing the asir (est) and associated 95% CI (lci, uci)

**References**

Boyle P, Parkin DM. Cancer registration: principles and methods. Statistical methods for registries. IARC Sci Publ. 1991;(95):126-58. PMID: 1894318.

**Examples**

```
ncan <- 1:10
py <- 101:110
pystd <- 10:1
ncan.min <- 5
age_standardized_incidence_rates(ncan, py, pystd, ncan.min)
age_standardized_incidence_rates(ncan, py, pystd, sum(ncan) + 1)
```

---

aggregated\_ageg\_name *Compute the aggregated age group names from a vector of more detailed age groups*

---

**Description**

Compute the aggregated age group names from a vector of more detailed age groups

**Usage**

```
aggregated_ageg_name(selected.ageg, ageg.sep = "_")
```

**Arguments**

selected.ageg	character, the fine grain age group vector
ageg.sep	character, the ageg group separator character

**Value**

character, the name of the aggregated age group

**Examples**

```
ageg.in <- c('15_19', '20_24', '25_29')
aggregated_ageg_name(ageg.in)
```

---

`chopped.vector.list`     *Pre-computed chopped combination for vectors size 1 to 18*

---

**Description**

This is a list containing all the possible combination of slices to chop vectors of size 1 to 18. It is useful to compute custom age group aggregation to ensure we are not disclosing age group with too few cancer cases.

**Usage**

```
chopped.vector.list
```

**Format**

A 18 item list:

each element is a matrix containing all the possible chop combinations to aggregate a vector of size n. ...

---

`chop_vector`     *Generate all the possible combinations of slices in a chopped vector*

---

**Description**

Generate all the possible combinations of slices in a chopped vector

**Usage**

```
chop_vector(vect.size = 3)
```

**Arguments**

`vect.size`     int, the size of the vector

**Value**

a matrix containing all the possible slices to chop a vector per line

**Examples**

```
chop_vector(3)
```

---

```
create_canradar_summary_file
```

*Compute summary statistics from 5 years age-group cancer registry data*

---

**Description**

Compute summary statistics from 5 years age-group cancer registry data

**Usage**

```
create_canradar_summary_file(  
  filename.in,  
  filename.out,  
  ncan.min = 5,  
  py.min = 100,  
  include.by.cob.stat = TRUE,  
  verbose = TRUE  
)
```

**Arguments**

filename.in	file path, the file containing the 5 years age counts of cancers stratified per cancer type, sex and country of birth
filename.out	file path, the file where summary .xlsx file will be save
ncan.min	integer, the minimum number of cancer per age group o be displayed
py.min	integer, the minimum number of person-years per age group to be considered during the aggregation step (default is 100)
include.by.cob.stat	logical, (TRUE by default) should the statistic per country-of-birth be computed and included in the output file.
verbose	logical, shall progress message be printed

**Value**

a .xlsx with all the summary statistics needed for Cancer RADAR project to be transmitted to project PIs.

### Examples

```
## Update file.in with the path to the input file containing your registry data
## (e.g. file.filled <- "cancerRADAR_input.xlsx")
file.in <- system.file("extdata", "ex_cancerRADAR_input_filled.xlsx", package = "cancerradarr")
file.out <- 'cancerRADAR_input.xlsx'
## for cancer radar data submission, we advise to use the parameter ncan.min = 5 and
## include.by.cob.stat = TRUE
create_canradar_summary_file(file.in, file.out, ncan.min = 20, include.by.cob.stat = FALSE)
## remove the file to pass package computation tests
unlink(file.out)
```

---

create\_registry\_input\_file

*Create a template file to be filled by cancer registry partners*

---

### Description

Create a template file to be filled by cancer registry partners

### Usage

```
create_registry_input_file(filename = "cancerRADAR_input.xlsx", verbose = TRUE)
```

### Arguments

filename	file path, the name of the template file to be created
verbose	logical, shall progress message be printed

### Value

a template .xlsx file is created on the hard drive.

### Examples

```
file.in <- 'input_file_test.xlsx'
create_registry_input_file(file.in)
## remove the file to pass package computation tests
unlink(file.in)
```

---

create\_static\_report    *Create a static report from cancer RADAR output file*

---

**Description**

Create a static report from cancer RADAR output file

**Usage**

```
create_static_report(filename.out = "")
```

**Arguments**

filename.out    file path, the path to a cancer RADAR output file  
This function will create a html report that could be useful to check the data that will be transmitted to IARC.

**Value**

nothing is returned, but a html file created with some summary statistics and graphs out of the file that should be transmitted with IARC

---

custom\_ageg\_aggregation  
*Smart aggregation of cancer cases per age group*

---

**Description**

Smart aggregation of cancer cases per age group

**Usage**

```
custom_ageg_aggregation(  
  dat,  
  ncan.min = 5,  
  py.min = 0,  
  add.total = FALSE,  
  ncan.lab = "ncan",  
  py.lab = "py"  
)
```

**Arguments**

dat	tibble, a single cancer/sex/country tibble containing cancer cases from a registry. It should contains the column ageg and ncan
ncan.min	integer, the minimal number of cancer in each category
py.min	integer, the minimal number of person-years in each category
add.total	logical, should the 'total' category added to the output dataset
ncan.lab	character, the column label where cancer cases are stored
py.lab	character, the column label where (optional) population at risk are stored

**Value**

aggregated dataset where all the age group contains at least ncan.min cancers cases

**Examples**

```
dat <-  
  dplyr::tribble(  
    ~ ageg, ~ ncan,  
    '00_04', 0,  
    '05_09', 0,  
    '10_14', 0,  
    '15_19', 0,  
    '20_24', 1,  
    '25_29', 2,  
    '30_34', 4,  
    '35_39', 5,  
    '40_44', 1,  
    '45_49', 10,  
    '50_54', 14,  
    '55_59', 1,  
    '60_64', 2,  
    '65_69', 2,  
    '70_74', 5,  
    '75_79', 1,  
    '80_84', 0,  
    '85', 0  
  )  
  
custom_ageg_aggregation(dat, 0)  
custom_ageg_aggregation(dat, 5)  
custom_ageg_aggregation(dat, 10)  
custom_ageg_aggregation(dat, 100)
```

---

`dat.aggr`*Geographical aggregation used for `cancerradarr`*

---

**Description**

In order to prevent loose of data in case of too low effective, several geographical aggregation can be considered. In this table are stored the different level of aggregation and the aggregation correspondence table considered.

**Usage**`dat.aggr`**Format**

A data frame with 250 rows and 5 columns:

**cob\_iso3** Country ISO3 code

**un\_region** UN region

**un\_subregion** UN subregion

**hdi\_cat** HDI 2023 category

**any\_migr** any migration background ...

**Details**

A multi-columns dataset containing all the countries of birth (as ISO3 code) and other geographical aggregation rules

---

`dat.asr.cat`*Burden of cancer aggregation category used for `cancerradarr`*

---

**Description**

A multi-columns dataset containing for all the countries of birth (as ISO3 code), sex and cancer type combinations the quartiles of cancer burden in country of origin burden. The quartiles (0%-24%, 25%-49%, 50%-74% and 75-100%) are based on the ASIR from GLOBOCAN 2022.

**Usage**`dat.asr.cat`

**Format**

A data frame with 2,220 rows and 5 columns:

**cob\_iso3** Country ISO3 code

**sex** targeted sex

**can** the cancer type

**asr** GLOBOCAN 2022 age-standardized cancer incidence rate

**asr\_rank\_cat** GLOBOCAN 2022 age-standardized cancer incidence rate quartile category ...

---

dat.cob

*Countries label and countries codes*

---

**Description**

A 2 column dataset containing all the countries of birth (with associated countries codes) included in Cancer RADAR project

**Usage**

dat.cob

**Format**

A data frame with 251 rows and 3 columns:

**cob\_label** Country name

**cob\_code** Country code

**cob\_iso3** Country ISO3 code (used as unique id) ...

---

diagnose\_input\_file

*Diagnose Input File for Data Quality Issues*

---

**Description**

This function checks the input file for common data quality issues that can cause errors during processing, including duplicate age groups and unexpected age group labels.

**Usage**

diagnose\_input\_file(filename.in)

**Arguments**

filename.in      file path, the file containing the 5 years age counts of cancers stratified per cancer type, sex and country of birth

**Value**

A list containing diagnostic information:

- `duplicate_age_groups`: Data frame of combinations with duplicate age groups
- `invalid_age_groups`: Character vector of invalid age group labels found
- `summary`: Overall summary of issues found

**Examples**

```
file.in <- system.file("extdata", "ex_cancerRADAR_input_filled.xlsx",
                      package = "cancerradarr")
diagnosis <- diagnose_input_file(file.in)
print(diagnosis$summary)
```

---

<code>globocan.2022.eu</code>	<i>European countries age-specific cancer burden from GLOBOCAN 2022</i>
-------------------------------	---

---

**Description**

A multi-columns dataset containing for all the European countries (UN definition) (as ISO3 code), sex and cancer type combinations the number of cases and population at risk estimated in GLOBOCAN 2022. This data are used in `cancerradarr` to compute the relative index on a standard reference population that could be more easily compared between registries. In addition to individual European countries, aggregated areas such as E27 (European Union 27 countries) and EUN (all the UN European countries) are stored in the dataset

**Usage**

```
globocan.2022.eu
```

**Format**

A data frame with 6,384 rows and 6 columns:

**cob\_iso3** Country ISO3 code

**sex** targeted sex

**ageg** targeted age group

**can** the cancer type

**ncanref** number of cancer cases estimated in GLOBOCAN 2022

**pyref** population at risk estimated in GLOBOCAN 2022 ...

**Source**

<https://gco.iarc.fr/today/en>

**References**

Bray F, Laversanne M, Sung H, Ferlay J, Siegel RL, Soerjomataram I, Jemal A. Global cancer statistics 2022: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J Clin.* 2024 May-Jun;74(3):229-263. doi: 10.3322/caac.21834. Epub 2024 Apr 4. PMID: 38572751.

---

incidence_rates	<i>Compute crude incidence rates</i>
-----------------	--------------------------------------

---

**Description**

Compute crude incidence rates

**Usage**

```
incidence_rates(ncan, py, ncan.min = 5, py.min = 0)
```

**Arguments**

ncan	integer, number of cancer
py	integer, number of person-year
ncan.min	integer, minimum number of observation required not to mask the CI's out
py.min	integer, minimum number of person-year required not to mask the CI's out

Crude incidence rates and associated 95% confidence interval are computing assuming a Poisson distribution and the exact method.

**Value**

a 3 column data.frame containing the crude incidence rate estimate (est) and associated 95% CI (lci, uci)

**References**

Boyle P, Parkin DM. Cancer registration: principles and methods. *Statistical methods for registries.* IARC Sci Publ. 1991;(95):126-58. PMID: 1894318.

**See Also**

[epitools::pois.exact\(\)](#)

**Examples**

```
ncan <- c(1, 10, 100)
py <- c(10, 100, 1000)
incidence_rates(ncan, py, 5, 500)
```

---

`incidence_rates_difference`*Compute incidence rates difference*

---

**Description**

Compute incidence rates difference

**Usage**

```
incidence_rates_difference(ncan, py, ncanref, pyref, ncan.min = 5, py.min = 0)
```

**Arguments**

<code>ncan</code>	integer, number of cancers in the population of interest
<code>py</code>	integer, person-year of the the population of interest
<code>ncanref</code>	integer, number of cancers in the reference population
<code>pyref</code>	integer, person-year of the the reference population
<code>ncan.min</code>	integer, minimum number of observation required not to mask the CI's out
<code>py.min</code>	integer, minimum person-year required not to mask the CI's out

Incidence rates differences and associated 95% confidence interval are computing assuming normal distribution of the differences..

**Value**

a 3 column data.frame containing the incidence rates difference (est) and associated 95% CI (lci, uci)

**Examples**

```
ncan <- 1:10  
py <- 101:110  
ncanref <- 41:50  
pyref <- 251:260  
ncan.min <- 5
```

```
incidence_rates_difference(ncan, py, ncanref, pyref, ncan.min)
```

---

incidence\_rates\_ratio *Compute incidence rates ratio*

---

**Description**

Compute incidence rates ratio

**Usage**

```
incidence_rates_ratio(ncan, py, ncanref, pyref, ncan.min = 5, py.min = 0)
```

**Arguments**

ncan	integer, number of cancers in the population of interest
py	integer, person-year of the the population of interest
ncanref	integer, number of cancers in the reference population
pyref	integer, person-year of the the reference population
ncan.min	integer, minimum number of observation required not to mask the CI's out
py.min	integer, minimum person-year required not to mask the CI's out

Incidence rates ratio and associated 95% confidence interval are computing assuming normal distribution of the ratios on the log scale.

**Value**

a 3 column data.frame containing the incidence rates ratio (est) and associated 95% CI (lci, uci)

**References**

Boyle P, Parkin DM. Cancer registration: principles and methods. Statistical methods for registries. IARC Sci Publ. 1991;(95):126-58. PMID: 1894318.

**Examples**

```
ncan <- 1:10
py <- 101:110
ncanref <- 41:50
pyref <- 251:260
ncan.min <- 5

incidence_rates_ratio(ncan, py, ncanref, pyref, ncan.min)
```

---

indirect\_proportional\_incidence\_ratio  
*Compute the indirect proportional incidence ratio (pir)*

---

**Description**

Compute the indirect proportional incidence ratio (pir)

**Usage**

```
indirect_proportional_incidence_ratio(  
  ncan,  
  ntot,  
  ncanref,  
  ntotref,  
  ncan.min = 5  
)
```

**Arguments**

ncan	integer, (age-specific) number of cancers in the population of interest
ntot	integer, (age-specific) total number of cancer the the population of interest
ncanref	integer, (age-specific) number of cancers in the reference population
ntotref	integer, (age-specific) total number of cancer the the reference of interest
ncan.min	integer, minimum number of observation required not to mask the CI's out

Indirect proportional incidence ratio and associated 95% confidence interval are computing assuming normal distribution of the pir on the log scale. pir is a summary statistics that should be computed per group of individuals providing age specific counts.

**Value**

a 1 line and 3 column data.frame containing the pir (est) and associated 95% CI (lci, uci)

**References**

Boyle P, Parkin DM. Cancer registration: principles and methods. Statistical methods for registries. IARC Sci Publ. 1991;(95):126-58. PMID: 1894318.

**Examples**

```
ncan <- 1:10  
ntot <- 11:20  
ncanref <- 41:50  
ntotref <- 251:260  
ncan.min <- 5
```

```
indirect_proportional_incidence_ratio(ncan, ntot, ncanref, ntotref, ncan.min)
indirect_proportional_incidence_ratio(ncan, ntot, ncanref, ntotref, sum(ncan) + 1)
```

---

```
indirect_standardized_incidence_ratio
    Compute indirect standardized incidence ratio (sir)
```

---

**Description**

Compute indirect standardized incidence ratio (sir)

**Usage**

```
indirect_standardized_incidence_ratio(
  ncan,
  py,
  ncanref,
  pyref,
  ncan.min = 5,
  py.min = 0
)
```

**Arguments**

<code>ncan</code>	integer, (age-specific) number of cancers in the population of interest
<code>py</code>	integer, (age-specific) person-year of the the population of interest
<code>ncanref</code>	integer, (age-specific) number of cancers in the reference population
<code>pyref</code>	integer, (age-specific) person-year of the the reference population
<code>ncan.min</code>	integer, minimum number of observation required not to mask the CI's out
<code>py.min</code>	integer, minimum number of person-years required not to mask the CI's out

Standardized incidence ratio (sir) and associated 95% confidence interval are computing assuming normal distribution of the pir on the log scale. sir is a summary statistics that should be computed per group of individuals providing age specific counts.

**Value**

a 1 line and 3 column data.frame containing the sir (est) and associated 95% CI (lci, uci)

**References**

Boyle P, Parkin DM. Cancer registration: principles and methods. Statistical methods for registries. IARC Sci Publ. 1991;(95):126-58. PMID: 1894318.

**Examples**

```
ncan <- 1:10
py <- 101:110
ncanref <- 41:50
pyref <- 251:260
ncan.min <- 5

indirect_standardized_incidence_ratio(ncan, py, ncanref, pyref, ncan.min)
indirect_standardized_incidence_ratio(ncan, py, ncanref, pyref, sum(ncan) + 1)
```

---

open\_canradar\_dictionary

*Open cancer RADAR output file dictionary*

---

**Description**

Calling this function will open the dictionary describing sheets and variables stored in the cancer summary file (output file generated by [create\\_canradar\\_summary\\_file](#) function). It could be useful for the cancer registries to check what kind of data they will be sharing. Note that a temporary copy of the dictionary is created on your hard drive to prevent from unwanted file modification.

**Usage**

```
open_canradar_dictionary()
```

**Value**

the path to a temporary file where cancer RADAR dictionary is stored

**Examples**

```
## Not run:
  open_canradar_dictionary()

## End(Not run)
```

---

proportional\_rates      *Compute proportional rates*

---

**Description**

Compute proportional rates

**Usage**

```
proportional_rates(ncan, ntot, ncan.min = 5)
```

**Arguments**

ncan	integer, number of cancer of interest
ntot	integer, overall number of cancer
ncan.min	integer, minimum number of observation required not to mask the CI's out

Proportional incidence rates and associated 95% confidence interval are computing assuming a Binomial distribution and the Clopper and Pearson (1934) procedure.

**Value**

a 3 column data.frame containing the proportional incidence rate estimate (est) and associated 95% CI (lci, uci)

**References**

C. J. CLOPPER, B.Sc., E. S. PEARSON, D.Sc., THE USE OF CONFIDENCE OR FIDUCIAL LIMITS ILLUSTRATED IN THE CASE OF THE BINOMIAL, *Biometrika*, Volume 26, Issue 4, December 1934, Pages 404–413, <https://doi.org/10.1093/biomet/26.4.404>

Boyle P, Parkin DM. Cancer registration: principles and methods. *Statistical methods for registries*. IARC Sci Publ. 1991;(95):126-58. PMID: 1894318.

**See Also**

`stats::binom.test()`

**Examples**

```
ncan <- c(1, 10, 100)
ntot <- c(10, 100, 1000)
proportional_rates(ncan, ntot, 5)
```

---

read\_cancerradar\_output\_01

*Read cancer registry summary statistics (non age-specific)*

---

**Description**

Read cancer registry summary statistics (non age-specific)

**Usage**

```
read_cancerradar_output_01(filename.out, aggr.level = "cob_iso3")
```

**Arguments**

`filename.out` file path, the path to a cancer RADAR output file

`aggr.level` character, the aggregation level to be considered. Should be one of `cob_iso3` (country of birth; smaller unit), `un_region` (UN region), `un_subregion` (UN sub-region), `hdi_cat` (HDI category), `asr_rank_cat` (cancer burden category in the country of birth), `any_migr` (any migration background)

**Value**

a tibble with 9 columns

- `reg_label`: factor, the chosen aggregation level id
- `sex`: character, male/female
- `ageg`: character, age group (here total)
- `can`: character, the cancer type
- `ref`: character, the reference population for relative index
- `index`: character, the type of index
- `est`: dbl, the index estimator
- `lci`: dbl, the index confidence interval lower bound
- `uci`: dbl, the index confidence interval upper bound

**Examples**

```
filename.out <- system.file('extdata/ex_cancerRADAR_output.xlsx', package = "cancerradarr")
dat.out <- read_cancerradar_output_01(filename.out, 'un_region')
head(dat.out)
```

---

read\_cancerradar\_output\_02

*Read cancer registry summary statistics (age-specific incidence rate and proportional rates)*

---

**Description**

Read cancer registry summary statistics (age-specific incidence rate and proportional rates)

**Usage**

```
read_cancerradar_output_02(filename.out, aggr.level = "cob_iso3")
```

**Arguments**

`filename.out` file path, the path to a cancer RADAR output file

`aggr.level` character, the aggregation level to be considered. Should be one of `cob_iso3` (country of birth; smaller unit), `un_region` (UN region), `un_subregion` (UN sub-region), `hdi_cat` (HDI category), `asr_rank_cat` (cancer burden category in the country of birth), `any_migr` (any migration background)

**Value**

a tibble with 11 columns

- `reg_label`: factor, the chosen aggregation level id
- `sex`: character, male/female
- `ageg`: character, age group (here total)
- `can`: character, the cancer type
- `index`: character, the type of index
- `est`: dbl, the index estimator
- `lci`: dbl, the index confidence interval lower bound
- `uci`: dbl, the index confidence interval upper bound
- `ageg_sta`: dbl, the age group starting age
- `ageg_sto`: dbl, the age group stopping age
- `ageg_mid`: dbl, the age group middle age

**Examples**

```
filename.out <- system.file('extdata/ex_cancerRADAR_output.xlsx', package = "cancerradarr")
dat.out <- read_cancerradar_output_02(filename.out, 'un_region')
head(dat.out)
```

---

read\_cancerradar\_output\_03

*Read cancer registry summary statistics (age-specific number of cancer cases)*

---

**Description**

Read cancer registry summary statistics (age-specific number of cancer cases)

**Usage**

```
read_cancerradar_output_03(filename.out, aggr.level = "cob_iso3")
```

## Arguments

filename.out file path, the path to a cancer RADAR output file

aggr.level character, the aggregation level to be considered. Should be one of cob\_iso3 (country of birth; smaller unit), un\_region (UN region), un\_subregion (UN sub-region), hdi\_cat (HDI category), asr\_rank\_cat (cancer burden category in the country of birth), any\_migr (any migration background)

## Value

a tibble with 10 columns

- reg\_label: factor, the chosen aggregation level id
- sex: character, male/female
- ageg: character, age group (here total)
- can: character, the cancer type
- index: character, the type of index
- ncan: dbl, the number of cancer
- py: dbl, the population size (if available)
- ageg\_sta: dbl, the age group starting age
- ageg\_sto: dbl, the age group stopping age
- ageg\_mid: dbl, the age group middle age

## Examples

```
filename.out <- system.file('extdata/ex_cancerRADAR_output.xlsx', package = "cancerradarr")
dat.out <- read_cancerradar_output_03(filename.out, 'any_migr')
head(dat.out)
```

---

run\_dynamic\_report *Create a dynamic report from cancer RADAR output file*

---

## Description

Create a dynamic report from cancer RADAR output file

## Usage

```
run_dynamic_report(filename.out = "")
```

## Arguments

filename.out file path, the path to a cancer RADAR output file

This function will open a shiny app where cancer registries can visually check the data they will be transmitted to IARC.

**Value**

nothing is returned

---

standardized\_incidence\_rate\_difference  
*Age-standardized incidence rates differences (asird)*

---

**Description**

Age-standardized incidence rates differences (asird)

**Usage**

```
standardized_incidence_rate_difference(  
  ncan,  
  py,  
  ncanref,  
  pyref,  
  pystd,  
  ncan.min = 5,  
  py.min = 0  
)
```

**Arguments**

ncan	integer, (age-specific) number of cancers in the population of interest
py	integer, (age-specific) person-year in the the population of interest
ncanref	integer, (age-specific) number of cancers in the reference population
pyref	integer, (age-specific) person-year in the the reference population
pystd	numeric, (age-specific) standard population person-years (e.g. standard world population)
ncan.min	integer, minimum number of observation required not to mask the CI's out
py.min	integer, minimum number of person-years required not to mask the CI's out

Age-standardized incidence rate difference (asird) is computed without confidence interval estimation for now. asird is a summary statistics that should be computed per group of individuals providing age specific counts.

**Value**

a 1 line and 3 column data.frame containing the pir (est) and associated 95% CI (lci, uci)

**References**

<https://www.hsph.harvard.edu/thegeocodingproject/analytic-methods/>

**Examples**

```

ncan <- 1:10
py <- 101:110
ncanref <- 41:50
pyref <- 251:260
pystd <- 10:1
ncan.min <- 5
standardized_incidence_rate_difference(ncan, py, ncanref, pyref, pystd, ncan.min)
standardized_incidence_rate_difference(ncan, py, ncanref, pyref, pystd, sum(ncan) + 1)

```

---

standardized\_incidence\_rate\_ratio

*Age-standardized incidence rates ratio (asirr)*

---

**Description**

Age-standardized incidence rates ratio (asirr)

**Usage**

```

standardized_incidence_rate_ratio(
  ncan,
  py,
  ncanref,
  pyref,
  pystd,
  ncan.min = 5,
  py.min = 0
)

```

**Arguments**

ncan	integer, (age-specific) number of cancers in the population of interest
py	integer, (age-specific) person-year in the the population of interest
ncanref	integer, (age-specific) number of cancers in the reference population
pyref	integer, (age-specific) person-year in the the reference population
pystd	numeric, (age-specific) standard population person-years (e.g. standard world population)
ncan.min	integer, minimum number of observation required not to mask the CI's out
py.min	integer, minimum person-year required not to mask the CI's out

Age-standardized incidence rate ratio (asirr) and associated 95% confidence interval are computing Armitage and Berry (1987) formula. asird is a summary statistics that should be computed per group of individuals providing age specific counts.

**Value**

a 1 line and 3 column data.frame containing the pir (est) and associated 95% CI (lci, uci)

**References**

Boyle P, Parkin DM. Cancer registration: principles and methods. Statistical methods for registries. IARC Sci Publ. 1991;(95):126-58. PMID: 1894318.

**Examples**

```
ncan <- 1:10
py <- 101:110
ncanref <- 41:50
pyref <- 251:260
pystd <- 10:1
ncan.min <- 5
standardized_incidence_rate_ratio(ncan, py, ncanref, pyref, pystd, ncan.min)
standardized_incidence_rate_ratio(ncan, py, ncanref, pyref, pystd, sum(ncan) + 1)
```

---

validate\_age\_groups      *Validate Age Group Labels*

---

**Description**

Checks that age group labels in a dataset match expected formats.

**Usage**

```
validate_age_groups(dat)
```

**Arguments**

dat                      A data frame containing an age column with age group labels.

**Details**

Expected age group labels are:

- Five-year age bands: '00\_04', '05\_09', ..., '80\_84'
- Open-ended group: '85'
- Special categories: 'total', 'DCO', 'MV'

**Value**

NULL (invisibly). The function is called for its side effect of stopping execution if invalid age groups are found.

**Examples**

```
dat <- data.frame(ageg = c("00_04", "05_09", "10_14"))
validate_age_groups(dat)
```

# Index

- \* **datasets**
  - chopped.vector.list, 4
  - dat.aggr, 9
  - dat.asr.cat, 9
  - dat.cob, 10
  - globocan.2022.eu, 11
- \* **data**
  - chopped.vector.list, 4
  - dat.aggr, 9
  - dat.asr.cat, 9
  - dat.cob, 10
  - globocan.2022.eu, 11
- \* **main**
  - create\_canradar\_summary\_file, 5
  - create\_registry\_input\_file, 6
  - run\_dynamic\_report, 21
- \* **summary-stat**
  - age\_standardized\_incidence\_rates, 2
  - incidence\_rates, 12
  - incidence\_rates\_difference, 13
  - incidence\_rates\_ratio, 14
  - indirect\_proportional\_incidence\_ratio, 15
  - indirect\_standardized\_incidence\_ratio, 16
  - proportional\_rates, 17
  - standardized\_incidence\_rate\_difference, 22
  - standardized\_incidence\_rate\_ratio, 23
- age\_standardized\_incidence\_rates, 2
- aggregated\_ageg\_name, 3
- chop\_vector, 4
- chopped.vector.list, 4
- create\_canradar\_summary\_file, 5, 17
- create\_registry\_input\_file, 6
- create\_static\_report, 7
- custom\_ageg\_aggregation, 7
- dat.aggr, 9
- dat.asr.cat, 9
- dat.cob, 10
- diagnose\_input\_file, 10
- epitools::pois.exact(), 12
- globocan.2022.eu, 11
- incidence\_rates, 12
- incidence\_rates\_difference, 13
- incidence\_rates\_ratio, 14
- indirect\_proportional\_incidence\_ratio, 15
- indirect\_standardized\_incidence\_ratio, 16
- open\_canradar\_dictionary, 17
- proportional\_rates, 17
- read\_cancerradar\_output\_01, 18
- read\_cancerradar\_output\_02, 19
- read\_cancerradar\_output\_03, 20
- run\_dynamic\_report, 21
- standardized\_incidence\_rate\_difference, 22
- standardized\_incidence\_rate\_ratio, 23
- stats::binom.test(), 18
- validate\_age\_groups, 24