CRATE Technical Details

2026-10-29

CRATE version 2.0.1

Installation

The user executes `source("crate-install.R")` once before any analysis script to install the R packages that CRATE depends on. `crate-install.R` performs the following steps:

- Detect operating system and R version. If the user is on Windows but not using the expected R version, they have to read and accept a message pointing that out before proceeding.
- If the user was on Windows and on the expected R version, we install the dependencies from binary archives (.zip). Otherwise we attempt to install from source code archives (.tar.gz) which is finicky on Windows. It can take a long time or even fail.
- Linux users always install from source code archives.
- The names of the R packages to install are detected from the contents of directory `dependencies/`.
- Directory `dependencies/` is a proper "repository" of R packages and we pass it to the `repos` argument of `install.packages`. R packages are not installed from anywhere else.

User settings

Overview

User settings are kept in /config/crate_user_settings.csv. The user fills into the column `setting_value` an appropriate value when there is no `setting_default` or if the `setting_default` is not right for them. The following settings have no default: qcs_dir_path, incidence_first_year, incidence_last_year, survival_follow_up_last_year, dataset_population_value_type.

The whole .csv except columns `setting_description` and `setting_value` (the former too long, the latter is for the user to fill):

setting_name	setting_example	setting_default
dataset_cancer_record_path	dataset.csv	dataset.csv
qcs_dir_path	C:/my/dir/qcs-x.x.x/	
java_command	C:/Program Files/OpenJDK/jdk-22.0.2/bin/java.exe	java
dataset_lifetable_path	my_lifetable.csv	lifetable.csv
dataset_lifetable_value_type	h	p
incidence_first_year	1985	
incidence_last_year	2020	
survival_follow_up_last_year	2020	
dataset_population_path	my_population.csv	population.csv
dataset_population_value_type	end	

Individual settings

dataset_cancer_record_path

Absolute or relative path to .csv containing cancer records as rows, as per the call for data for the 'incidence' dataset.

`CRATE` calls `QCS` on the dataset in this location.

qcs_dir_path

Absolute or relative path to QCS directory. This directory must contain the main executable that looks like jrc-qcs-x.x.x.jar.

`CRATE` calls the `QCS` found in this location. This directory must contain the main .jar file of `QCS`. The version of `QCS` is required to be >= 2.4.1.

java_command

Name of command or absolute path to Java executable. If Java has been installed and the java executable is included via the system PATH environment variable, then the simple command line command `java` will be available and the default will work directly. If you need to e.g. use a different version of Java, you can supply the path to its executable here.

When `CRATE` calls `QCS`, a Java programme, this is used as the Java command.

dataset_lifetable_path

Relative or absolute path to lifetable file. Must be csv.

The lifetable file is read from this location.

dataset_lifetable_value_type

Indicates whether the lifetable contains probabilities (p) or hazard rates (h).

The lifetable file must contain either death probabilities `dataset_lifetable_value_type = p` or mortality rates `dataset_lifetable_value_type = h`. This only affects preprocessing: we convert the death probabilities to the mortality rates expected by the estimation software. As a special consideration we allow a death probability value of one, which may happen in small strata. However, this is converted to 0.95 because a death probability of one implies an infinite mortality which cannot be used. CRATE warns when this happens into the console and log.

incidence_first_year

First year covered sufficiently completely by incidence registration.

- Used to determine the "soft" limits for periods of incidence so that no period has too little data. See the documentation for column `POI`.
- Used to determine which years to use when tabulating by `YOI`. The first year is `incidence_first_year` from `crate_user_settings` or `survival_incidence_first_year` from `crate_global_settings`, whichever is larger. The last year is `incidence_last_year` from `crate_user_settings` or `survival_incidence_last_year` from `crate_global_settings`, whichever is smaller.
- In all survival analysies except period analysis, we use as the hard lower limit on follow-up along the calendar time scale `incidence_first_year` from `crate_user_settings` or `survival_incidence_first_year` from `crate_global_settings`, whichever is greater. This hard lower limit is unlikely to have any effect but is required because the software needs also a lower limit when we use an upper limit. (This hard lower limit could also be e.g. the year 1000, but using this approach reminds us what the real limit is.)

- Used to filter out records before survival estimation. We retain only cases where `incidence_first_year <= YOI <= incidence_last_year`.
- Used to exclude cases in prevalence analysis such that we only keep cases where `incidence_first_year <= YOI <= incidence_last_year`.

incidence_last_year

First year covered sufficiently completely by incidence registration.

- Used to determine the "soft" limits for periods of incidence so that no period has too little data. See the documentation for column `POI`.
- Used to determine which years to use when tabulating by `YOI`. The first year is `incidence_first_year` from `crate_user_settings` or `survival_incidence_first_year` from `crate_global_settings`, whichever is larger. The last year is `incidence_last_year` from `crate_user_settings` or `survival_incidence_last_year` from `crate_global_settings`, whichever is smaller.
- Used to filter out records before survival estimation. We retain only cases where `incidence_first_year <= YOI <= incidence_last_year`.
- Used to exclude cases in prevalence analysis such that we only keep cases where `incidence_first_year <= YOI <= incidence_last_year`.

survival_follow_up_last_year

Last year of survival follow-up. E.g. with 2020 this means that follow-up reaches 2020-12-31.

- In all survival analysies except period analysis, we use as the hard upper limit on follow-up along the calendar time scale `survival_follow_up_last_year` from `crate_user_settings` or `survival_follow_up_last_year` from `crate_global_settings`, whichever is earlier in time. This guarantees that all records obey both intended upper limits of follow-up in calendar time.

dataset_population_path

Relative or absolute path to population file. Must be csv.

The population dataset is read from this path.

dataset_population_value_type

Indicates whether the population file contains the 'mid-year' (mid), start of the year (start), or end of the year population (end).

Setting `dataset_population_value_type` affects how the column `Number_of_residents` is preprocessed. We use the end-of-the-year population size for computing prevalence proportions, so the following actions are taken depending on this setting:

- `end`: Use the values as-is.
- `start`: If possible, turn these into end-of-year values by taking the value from the next year within each stratum. This is not possible if the population dataset does not have any data after the prevalence observation year — see `prevalence_observation_year` in `crate_global_settings`.
- `mid`: Use the values as they are, but emit a message into the log and console that the value type was not ideal. One could take the mean of two subsequent years to get something closer to the year-end value, but it is perhaps a hacky solution and not implemented.

Global settings

Overview

The user must not change any of these when participating in an ECIS data call.

Outside a data call one is free to experiment.

Contents:

setting_name	setting_value
survival_incidence_first_year	1981
survival_incidence_last_year	2024
survival_follow_up_last_year	2024
survival_incidence_period_width	5
survival_period_analysis_width	4
survival_max_age_year	89
prevalence_observation_year	2020
prevalence_interval_limits	c(0:5, 10, 15, 20, 25, Inf)

Individual settings

survival_incidence_first_year

- Used to determine the hard limits for periods of incidence. See the documentation for column `POI`.
- Used to determine which years to use when tabulating by `YOI`. The first year is `incidence_first_year` from `crate_user_settings` or `survival_incidence_first_year` from `crate_global_settings`, whichever is larger. The last year is `incidence_last_year` from `crate_user_settings` or `survival_incidence_last_year` from `crate_global_settings`, whichever is smaller.
- In all survival analysies except period analysis, we use as the hard lower limit on follow-up along the calendar time scale `incidence_first_year` from `crate_user_settings` or `survival_incidence_first_year` from `crate_global_settings`, whichever is greater. This hard lower limit is unlikely to have any effect but is required because the software needs also a lower limit when we use an upper limit. (This hard lower limit could also be e.g. the year 1000, but using this approach reminds us what the real limit is.)
- Used to filter out records before survival estimation. We retain only cases where `survival_incidence_first_year <= YOI <= survival_incidence_last_year`.

survival_incidence_last_year

- Used to determine the hard limits for periods of incidence. See the documentation for column `POI`.
- Used to determine which years to use when tabulating by `YOI`. The first year is `incidence_first_year` from `crate_user_settings` or `survival_incidence_first_year` from `crate_global_settings`, whichever is larger. The last year is `incidence_last_year` from `crate_user_settings` or `survival_incidence_last_year` from `crate_global_settings`, whichever is smaller.
- Used to filter out records before survival estimation. We retain only cases where `survival_incidence_first_year <= YOI <= survival_incidence_last_year`.

survival_follow_up_last_year

- Used to determine the last year used in period analysis in survival estimation.
- Compared against the maximum year in the `lifetable` dataset. See the documentation for the `lifetable` dataset.
- In all survival analysies except period analysis, we use as the hard upper limit on follow-up along the calendar time scale `survival_follow_up_last_year` from `crate_user_settings` or `survival_follow_up_last_year` from `crate_global_settings`, whichever is earlier in time. This guarantees that all records obey both intended upper limits of follow-up in calendar time.
- Used to determine the global roof in follow-up along the calendar time scale in the definition of `flag_nd_qi_4`. See that column's definition for details.

survival_incidence_period_width

- Used to determine how periods of incidence are created. See the documentation for column `POI`.

survival_period_analysis_width

- Used to determine the width of the period in period analysis in survival estimation.

survival_max_age_year

Used to determine the upper limit of the last age group in column `ECIS_AG_SURV`.

prevalence_observation_year

Determines the sole prevalence observation time point. We estimate prevalence at the end of this year.

Because we only know the year and month of diagnosis and not the exact date, but we do use the exact days of follow-up from `Surv_time`, we use in calculations the date `yyyy-12-16` as the prevalence observation time point. This date is guaranteed to be exceeded by those diagnosed (in any year and month) after the middle of the month and whose follow-up extends to the end of the same year we calculate prevalence for. For instance, being actually diagnosed in 2020-09-30 and having `Surv_time = 1553` to the actual end of follow-up in 2024-12-31 would be fine if we knew the exact date, but we are forced to assume 2020-09-15, and 1553 days from that is 2024-12-16. So if we were to use 2024-12-31 as the prevalence observation date, all records of this kind would be artificially left out. (The alternative approach is to abandon the use of `Surv_time` at least in prevalence, but that seemed like a lot of work and a source of new uncertainties.)

- Used in the creation of table `prevalence_latest`.
- Used in the creation of table `prevalence_latest_by_ts_age`.
- Used to subset data in prevalence analysis. Cancers diagnosed after this year are excluded. E.g. `YOI = 2025` and `prevalence_observation_year = 2024` means that record is excluded.

prevalence_interval_limits

Used to determine the limits of follow-up intervals in all prevalence estimation. This string must be evaluatable R code. E.g. `c(0, 5, 10)` causes 5-year and 10-year prevalence to be estimated. The zero is optional and is included always anyway.

Datasets

Incidence

We create new columns into the `incidence` dataset for use in survival and prevalence analysis.

PAT

`PAT` is marked missing (`NA`) if `ERR-100` is found in `FLAG_SURV`.

TUM

`TUM` is marked missing (`NA`) if `ERR-101` is found in `FLAG_SURV`.

TOPO

`TOPO` is marked missing (`NA`) if `ERR-104` is found in `FLAG_SURV`.

MORPHO

`MORPHO` is marked missing (`NA`) if `ERR-105` is found in `FLAG_SURV`.

BEH

`BEH` is marked missing (`NA`) if `ERR-106` is found in `FLAG_SURV`.

SEX

`SEX` is marked missing (`NA`) if `ERR-102` is found in `FLAG_SURV`. Also if value is not 1 or 2.

MOB

`MOB` is marked missing (`NA`) if `WARN-201` is found in `FLAG_SURV`. Also if value is 99.

YOB

`YOB` is marked missing (`NA`) if `WARN-200` is found in `FLAG_SURV`. Also if value is 9999.

MOI

`MOI` will be set to 6 if `WARN-203` is found in `FLAG_SURV`. Also if value is 99.

YOI

`YOI` is marked missing (`NA`) if `ERR-103` is found in `FLAG_SURV`. Also if value does not pass this condition: `survival_incidence_first_year <= YOI <= survival_incidence_last_year` from `crate_global_settings`.

SURV_TIME

`SURV_TIME` is marked missing (`NA`) if `WARN-208` is found in `FLAG_SURV`. Also if value is 99999.

VIT_STAT

`VIT_STAT` is marked missing (`NA`) if `ERR-108` is found in `FLAG_SURV`. Value 9 is *not* converted to `NA`.

MOF

`MOF` is marked missing (`NA`) if `WARN-206` is found in `FLAG_SURV`. Also if value is 99.

YOF

YOF is marked missing (NA) if WARN-205 is found in FLAG_SURV. Also if value is 9999.

DOI

Date of incidence is assumed to be on the 15th of the year-month taken from YOI and MOI.

DOI will be missing if YOI is missing.

DOF

Date of end of follow-up is assumed to be on the 16th of the year-month taken from YOF and MOF. This will be missing if either of those two is missing.

DOB

Date of birth is assumed to be on the 15th of the year-month taken from YOB and MOB. This will be missing if either of those two is missing.

lex.dur

This is the follow-up time in fractional years and used in the actual analyses. We take it as SURV_TIME / 365.24 at first. If SURV_TIME is missing and DOI and DOF are not, then we impute this as the difference between those two dates.

ts_fut

This indicates the follow-up time at the start follow-up, which is always zero. This column is required by the survival estimation software.

ts_age

This is the age at diagnosis in fractional years, defined as the difference between DOI and DOB initially. If either of those is missing and AGE is not, then we impute this with AGE + 0.5.

ts_cal

The calendar time in fractional years at diagnosis is simply DOI turned into fractional years.

POI

POI, period of incidence, is formed using YOI as follows:

- We use a period width from crate_global_settings setting survival_incidence_period_width.
- The latest incidence period excludes the years used separately in period analysis. E.g. with a period width of four and if we compute period analysis for 2021-2024 then the latest incidence period would be 2017-2020.
- Besides the exclusion, we initially create periods starting from survival_incidence_last_year and going back as far as survival_incidence_first_year from crate_global_settings allows. E.g. with survival_incidence_first_year = 1900, survival_incidence_last_year = 2024, and width 4 we initially have incidence periods 1901-1904, 1905-1908, ..., 2013-2016, 2017-2020.
- Naturally we cannot make use of periods where there is no data, or not enough data. To this end we compute how many years are missing from a period based on incidence_first_year and incidence_last_year from crate_user_settings. For instance, with incidence_first_year = 2007 the period 2005-2008 would have two missing years.
- We allow at most a 40 % deficit in a period, so for instance the following are excluded: missing 2 out of 3, 2 out of 4, 3 out of 5. Missing only 1 out of 3, 1 out of 4, or 2 out of 5 does not lead to exclusion.

- And so we end up with only those periods from the initial ones which have sufficient data. Good `POI`. This of course affects everything where `POI` is used. If something is supposed to be estimated by `POI` then we estimate using only the periods existing in `POI`.
- For instance based on `crate_global_settings` we might create the periods 1981-1985, ... 2016-2020. But `incidence_first_year = 1984` from `crate_user_settings` would cause `POI` to only have the periods 1986-1990, ..., 2016-2020. Therefore, all tables of estimates stratified by `POI` would in this case not include the period 1981-1985.

ECIS_POI

This column contains only the single period 2000-2007 and data from other years of incidence (`YOI`) have value `NA`.

ECIS_AG_SURV

Created using column `AGE`. We use the age groups 14-44, 45-54, 55-64, 65-74, and as the last `75-survival_max_age_year` from `crate_global_settings`. E.g. `75-89`.

FLAG_SURV

Produced as follows:

- First delete the `temp/` subdir in `qcs_dir_path` to ensure we cannot read the wrong result set from there (from some previous run).
- Run QCS in the background.
- Read the QCS results for the incidence dataset. To be exact, we read the `qcs_rule_output_x.csv` table only.
- We detect the codes to retain using the regex `((ERR)|(WARN))[-][0-9]+$`. This causes all codes that look like `ERR-X` or `WARN-X` to be retained, where `X` is a number of any length.
- Read columns `PAT` and `TUM` from the incidence dataset. These are used to join the QCS output to our analysis dataset.
- Create an aggregate column of the error and warning codes where each element is a string that separates multiple codes with ",". E.g. one record may have the value "ERR-X, WARN-X".
- Mark elements without any error and warning codes as "ok".
- Perform a join with the incidence dataset by the columns `PAT` and `TUM`.

FLAG_PREV

Taken as an unmodified copy of `FLAG_SURV`.

AG_QI

This age group column for quality indicators contains the age groups 0-14 and 15+.

flag_nd_qi_1

Always `TRUE`.

flag_n_qi_1

`TRUE` when all of the following are met:

- `flag_nd_qi_1 == TRUE`
- At least one `ERR-X` code is found in `FLAG_SURV`

flag_nd_qi_1a

Identical to `flag_nd_qi_1`.

flag__n__qi__1a

TRUE when all of the following are met:

- flag_nd_qi_1a == TRUE
- VIT_STAT == 1
- ERR-116 is found in FLAG_SURV

flag__nd__qi__1b

Identical to flag_nd_qi_1.

flag__n__qi__1b

TRUE when all of the following are met:

- flag_nd_qi_1b is TRUE
- VIT_STAT == 2
- BOD != 0 OR AUTOPSY != 1
- ERR-116 is found in FLAG_SURV

flag__nd__qi__2

Identical to flag_nd_qi_1.

flag__n__qi__2

TRUE when all of the following are met:

- flag_nd_qi_2 is TRUE
- BOD == 0
- No ERR-X code is found in FLAG_SURV

flag__nd__qi__3

Identical to nd_qi_1.

flag__n__qi__3

TRUE when all of the following are met:

- flag_nd_qi_3 is TRUE
- AUTOPSY == 1
- BOD != 0
- No ERR-X code is found in FLAG_SURV

flag__nd__qi__4

TRUE when all of the following are met:

- VIT_STAT == 1
- Time of diagnosis is such that there could have been at least four years of follow-up. I.e. the case could have contributed to the survival estimation in the 5th year of follow-up.

In the code this test of potential follow-up is accomplished by comparing the fractional year value of the time of diagnosis (e.g. 2020-10-25 is approximately 2020.789) to the fractional year value of the global roof value, which is crate_global_settings setting survival_follow_up_last_year plus one. E.g. with survival_follow_up_last_year = 2024 the distance between the global roof 2025.000 and 2020-10-25 in fractional years is approximately 4.211. When this distance is at least 4.0, the case could have contributed to the survival estimation in the 5th year of follow-up. Whether they factually do or not is not tested for here.

If more than one period is used for period analysis then this will be broken. E.g. the upper limit of calendar time for period 2017-2020 would be 2020 + 1.

flag__n__qi__4

`TRUE` when all of the following are met:

- `flag_nd_qi_4 == TRUE`
- Observed follow-up less than 4 years. Such a record does not contribute to the survival estimation in the 5th year of follow-up.

flag__nd__qi__6

Identical to `flag_nd_qi_1`.

flag__n__qi__6

`TRUE` when all of the following are met:

- `flag_nd_qi_6 == TRUE`
- Follow-up time less than a year
- `VIT_STAT == 2`

Population

CRATE preprocesses the `population` dataset as follows:

- The dataset is read from the path given in `dataset_population_path` in `crate_user_settings`.
- For the sake of consistency, replace any spaces in column names with the underscore "_". E.g. `Age range` -> `Age_range`. Make all column names start with a capital letter and all other letters lowercase.
- Remove column `Geo_label` if there. The column `Geo_code` is sufficient.
- Remove column `Geo_code` if all its values are missing (`NA`).
- Uppercase the following column names found in the dataset: `Registry_id`, `Geo_code`, `Sex`. These are in uppercase in the incidence dataset within CRATE.
- If the dataset contains the column `Age_unit` or `Annual_Age`, we harmonise it to look the same as if `Age_range` had been supplied. E.g. from value `1` to value `1-1`. The column is renamed to `Age_range`. This is done because ranges are the most general expression of age.
- If `Age_range` was supplied, each value can be one of the following (you can mix different types in the same column):
  - A single age year, e.g. `0`.
  - A range like `1-4`.
  - A range without an upper limit like `90+`.
- This column must not contain age groups as age group IDs such as 1 for `0-4`, 2 for `5-9`, and so on. To be clear, the programme *does* allow one-year age groups such as 0, 1, 2, ..., 109, 110.
- For harmonisation of the current `Age_range` column, the lower and upper limit of each age group is extracted from `Age_range`. For a single-year value such as `0` this means that the age group comprises set of age years starting from and ending in zero.
- We always make the upper limit of the last `Age_range` 1000, i.e. no upper limit for the last value of `Age_range`.
- From the lower and upper limits we determine which age groups belong to which of the 18 age groups we harmonise into, 0-4, 5-9, ..., 80-84, 85+.
- The column containing values is assumed to be named `Number_of_residents`. Other columns are assumed to be stratifying columns. Therefore, those running CRATE for multiple registries in one go can include the column `REGISTRY_ID` also in the `population` dataset.
- This includes re-summing the `Number_of_residents` values so that we end up with 18 age groups for every combination of the other stratifying columns.

There is further processing of the value column depending on setting `dataset_population_value_type` from `crate_user_settings`.

Setting `dataset_population_value_type` affects how the column `Number_of_residents` is preprocessed. We use the end-of-the-year population size for computing prevalence proportions, so the following actions are taken depending on this setting:

- `end`: Use the values as-is.
- `start`: If possible, turn these into end-of-year values by taking the value from the next year within each stratum. This is not possible if the population dataset does not have any data after the prevalence observation year — see `prevalence_observation_year` in `crate_global_settings`.
- `mid`: Use the values as they are, but emit a message into the log and console that the value type was not ideal. One could take the mean of two subsequent years to get something closer to the year-end value, but it is perhaps a hacky solution and not implemented.

The harmonised dataset is used in estimation as follows:

- Passed to `popEpi::prev_lexis`. This causes the computation of (possibly age adjusted) prevalence proportions.

Lifetable

The `lifetable` dataset is preprocessed as follows:

- The dataset is read from path `dataset_lifetable_path` from `crate_user_settings`.
- For the sake of consistency, replace any spaces in column names with the underscore "_". E.g. `Age range` -> `Age_range`. Make all column names start with a capital letter and all other letters lowercase.
- Remove column `Geo_label` if there. The column `Geo_code` is sufficient.
- Remove column `Geo_code` if all its values are missing (`NA`).
- Uppercase the following column names found in the dataset: `Registry_id`, `Geo_code`, `Sex`. These are in uppercase in the incidence dataset within CRATE.
- If the dataset contains the column `Age_unit` or `Annual_Age`, we harmonise it to look the same as if `Age_range` had been supplied. E.g. from value `1` to value `1-1`. The column is renamed to `Age_range`. This is done because ranges are the most general expression of age.
- If `Age_range` was supplied, each value can be one of the following (you can mix different types in the same column):
  - A single age year, e.g. `0`.
  - A range like `1-4`.
  - A range without an upper limit like `90+`.
- This column must not contain age groups as age group IDs such as 1 for `0-4`, 2 for `5-9`, and so on. To be clear, the programme *does* allow one-year age groups such as 0, 1, 2, ..., 109, 110.
- For harmonisation of the current `Age_range` column, the lower and upper limit of each age group is extracted from `Age_range`. For a single-year value such as `0` this means that the age group comprises set of age years starting from and ending in zero.
- We always make the upper limit of the last `Age_range` 1000, i.e. no upper limit for the last value of `Age_range`.
- From the lower and upper limits we produce the kind of age groups expected by the survival estimation software, which look like e.g. `[0, 5[`.
- Detect the name of the value column, either `All_causes_death_probability` or `All_causes_death_rate`.

We preprocess the value column according to the setting `dataset_lifetable_value_type` from `crate_user_settings`.

The lifetable file must contain either death probabilities `dataset_lifetable_value_type = p` or mortality rates `dataset_lifetable_value_type = h`. This only affects preprocessing: we convert the death probabilities to the mortality rates expected by the estimation software. As a special consideration we allow a death probability value of one, which may happen in small strata. However, this is converted to 0.95 because a death probability of one implies an infinite mortality which cannot be used. CRATE warns when this happens into the console and log.

- Define as the stratifying columns all other columns except the value column. Therefore, those running CRATE for multiple registries may include the column `REGISTRY_ID` also in the `lifetable` dataset.
- For survival estimation to work correctly, the `lifetable` dataset must contain data for the year `survival_follow_up_last_year` from `crate_global_settings`. For convenience we repeat the last year in the `lifetable` dataset to fill the gap to the `survival_follow_up_last_year` if such a gap exists.
- If at most two years were added in this way, we inform the user of this but progress normally.
- If more than two years were added, we emit both a message and a warning (see `?warning` in R). But the run still proceeds.

Dataset `lifetable` has the following uses:

- Used in survival estimation for producing e.g. Pohar Perme estimates.

Survival and prevalence quality indicators

Methodology

QIs are produced as follows:

- QI 5 is produced separately in prevalence estimation.
- We use the following subsets based on the QI we are computing:
- 1, 1a, 1b, 2, 3: The exact same non-quality conditions as always used in survival estimation:
  - Age in whole years is at most `survival_max_age_year` taken from `crate_global_settings`
  - `survival_incidence_first_year <= YOI <= survival_incidence_last_year` taken from `crate_global_settings`
  - `incidence_first_year <= YOI <= incidence_last_year` taken from `crate_user_settings`.
- 4 and 6: The exact same conditions always used in survival estimation:
  - Age in whole years is at most `survival_max_age_year` taken from `crate_global_settings`
  - `survival_incidence_first_year <= YOI <= survival_incidence_last_year` taken from `crate_global_settings`
  - `incidence_first_year <= YOI <= incidence_last_year` taken from `crate_user_settings`.
  - follow-up time is > 0
  - `FLAG_SURV` contains no `ERR-X` codes
  - `SEX` is 1 or 2
  - `BOD != 0`
  - `AUTOPSY != 1`
- Each quality indicator has a numerator and a denominator. These both have a corresponding flag column which we create into the incidence dataset. We simply sum over these flag columns which one can view as count columns containing only a zero or a one for a record. See e.g. the definition of column `flag_n_qi_1`.
- For the following QIs, only compute statistics for `ENTITY_ID = 1`: 1, 1a, 1b, 3, 4, 6. These QIs were only specified for that one cancer entity, and so we set the values for the rest as missing. If the quality indicator 2 was not requested then we drop those rows from the result table altogether.
- We set `ENTITY_ID = 1` for cases where `ENTITY_ID = 0` and any one of `TOPO`, `MORPHO`, `BEH` is missing. This is only for the duration of computing QIs and affects nothing else. We want to catch also major errors where one of those columns is missing within the `ENTITY_ID = 1` statistics.
- Compute a table of quality indicator denominators and numerators separately for each cancer entity. They are combined into one big table with `ENTITY_ID` as a stratifying column.
- We take the first record per person within the subset based on entity and exclusions and use only those for computing the quality indicators. E.g. if a person has three records but the first one is not in the subset, then only that person's second record is used in the computation.
- Compute the proportion from the denominator and numerator for each QI. These columns appear as `p_*` columns, e.g. `p_qi_1`.

Tables of quality indicators

- All tables are stratified by `ENTITY_ID` and `AG_QI`.

Possible columns in each table of quality indicators:

- Stratifying columns such as `ENTITY_ID`.
- The columns indicating intervals along some Lexis time scale, e.g. calendar time. For calendar time these look like `ts_cal_start`, `ts_cal_stop`.
- `n_qi_x`, e.g. `n_qi_1` for the numerator of the quality indicator.
- `nd_qi_x`, e.g. `nd_qi_1` for the denominator of the quality indicator.
- `p_qi_x`, e.g. `p_qi_1` for the proportion computed from the two.

`quality_indicators_latest_period`

- This table simply contains the quality indicators for the same period analysis period as the corresponding table of survival estimates, `survival_latest_period`. To this end we only use records who have some follow-up in the 0-5 interval within the period of interest. E.g. for period 2021-2024, if a record has reached 5.5 years of follow-up on the first day of 2021, then that record is excluded.
- The period is indicated by the columns `ts_cal_start` and `ts_cal_stop`.
- QIs 1, 1a, 1b, 3, and 4.

`quality_indicators_by_POI`

- Stratified also by `POI`.
- QIs 1, 1a, 1b, 3, and 4.

`quality_indicators_by_YOI`

- Stratified also by `YOI`.
- QIs 2 and 6.

Survival

Survival estimation

What follows is a general description of survival estimation in CRATE. How individual tables are produced may have exceptions or additions to what what is described here — see section **Tables of survival estimates**.

- Only keep cancer records where
  - Age in whole years is at most `survival_max_age_year` taken from `crate_global_settings`
  - `survival_incidence_first_year <= YOI <= survival_incidence_last_year` taken from `crate_global_settings`
  - `incidence_first_year <= YOI <= incidence_last_year` taken from `crate_user_settings`.
  - follow-up time is > 0
  - `FLAG_SURV` contains no `ERR-X` codes
  - `SEX` is 1 or 2
  - `BOD != 0`
  - `AUTOPSY != 1`
- The following steps are performed separately for each cancer entity.
- Take the first cancer record of the given entity for each patient if there are multiple. This occurs *after* the filtering out above. Therefore, if a person has e.g. three cancer diagnoses of the same type and but the first was filtered out, then only the second diagnosis is used in survival estimation.
- Perform the computationally heavy step, split-merge-aggregate. The follow-up of the cancer records is split into intervals of survival (e.g. monthly intervals). If the lifetable needs to be used (because we are estimating net/relative survival), the contents of the lifetable are merged to those intervals. Then summary statistics such as the time at risk, `t_at_risk` are computed for the intervals and aggregated. This results in a table of summary statistics by the strata we are interested in (and if age-adjusting was requested, by age group).
- If age-adjusting was requested, the table produced by the above step contains all the age groups allowed for by the ICSS standard, which at the time of writing this are 5-year age groups starting from age group 15-19 and ending with 85+. But because Brenner weighting requires every adjusting stratum to contain at least one cancer record, we re-aggregate the current table of summary statistics so as to produce possibly larger age groups which all contain at least one cancer record. E.g. if the first age group 15-19 contained no cancer record but all other age groups did, then we arrive at the first age group being 15-24 and the others remain untouched. This approach allows for robust age-adjusting (and as a side effect with the highest possible fidelity, though this is unlikely to matter very much in practice).
- For technical reasons we currently also require that every aggregate age group has a survival curve where every survival interval has `t_at_risk > 0`. This is done to ensure that the survival interval aggregation step below does not aggregate survival intervals differently for different age groups. E.g. if age group `15-49` has no follow-up after 3.5 years but other age groups do have, then age group might result in the last survival intervals being `]4, 5]` but for other age groups it might be `]4.916, 5.0]`. These survival intervals cannot be aggregated when we compute the Brenner weighted statistics. Long story short, the reason for this step is purely because we have not implemented anything fancier. So the exact condition for aggregating age groups is that there is at least one person in follow-up at `T=0` and that `t_at_risk > 0` in every survival interval.
- Perform another re-aggregation step regardless of whether we are adjusting for age or not: It is possible that smaller cancer entities have such sparse data that the survival intervals (at the time of writing this, monthly intervals) may not all have data. This prevents survival estimation in such intervals of course. For this reason we combine these empty intervals until every interval has `t_at_risk > 0`, with the restriction that we do not combine intervals across the even years. For instance if the 24th monthly interval has zero follow-up, it gets combined with the the 23rd interval. (The 24th interval ends at T=2

so it could not be combined with the 25th interval.) This step handles automatically these problematic edge cases caused by sparse data (and typically this happens in period analysis and not otherwise). Note that it is theoretically possible that an entire year of follow-up is empty, in which case no survival estimates can be produced even with our re-aggregation trickery.

- Compute the survival estimates using `popEpi::surv_estimate`. If age-adjusting was requested then the weights are applied at this point. See section **Age-adjusting** for details. This also produces confidence intervals, for which we always use the log transform.
- Retain only estimates at the even years and aggregate all summary statistics such as `t_at_risk` over the months within the year. This results in a table with annual estimates and summary statistics.

Age-adjusting

The Brenner weights are determined as follows:

- Standard weights are fetched for the current `entity_id`. Different cancer types can have different standard weights. E.g. these standard weights:

ag_icss	weight
15 - 19	0.00164
20 - 24	0.00215
25 - 29	0.00416
30 - 34	0.00842
35 - 39	0.01874
40 - 44	0.03489
45 - 49	0.04906
50 - 54	0.07094
55 - 59	0.09641
60 - 64	0.13359
65 - 69	0.14234
70 - 74	0.14766
75 - 79	0.14131
80 - 84	0.09347
85 - Inf	0.05522

- The observed weights by age group — patient numbers — are computed for each output stratum separately using only those cancer records which we allow to be used in survival analysis and which were diagnosed in the period of interest — e.g. in period analysis for 2020-2023 we would only use cases diagnosed in 2020-2023. One output stratum can be female breast cancer patients in a certain period. Therefore although the standard weights can be the same for many strata, the Brenner weights are likely always somewhat different. E.g. we might have these observed weights:

SEX	ag_icss	weight
1	15 - 19	0.0035194
1	20 - 24	0.0048839
1	25 - 29	0.0068548
1	30 - 34	0.0088307
1	35 - 39	0.0123936
1	40 - 44	0.0190597
1	45 - 49	0.0326613
1	50 - 54	0.0581662
1	55 - 59	0.0973905
1	60 - 64	0.1399647
1	65 - 69	0.1685793
1	70 - 74	0.1703019
1	75 - 79	0.1369002
1	80 - 84	0.0841388
1	85 - Inf	0.0488469
2	15 - 19	0.0034047
2	20 - 24	0.0050961
2	25 - 29	0.0082776
2	30 - 34	0.0140885
2	35 - 39	0.0234333
2	40 - 44	0.0383424

SEX	ag_icss	weight
2	45 - 49	0.0608126
2	50 - 54	0.0799317
2	55 - 59	0.0962285
2	60 - 64	0.1175660
2	65 - 69	0.1280586
2	70 - 74	0.1298138
2	75 - 79	0.1189789
2	80 - 84	0.0923624
2	85 - Inf	0.0771244

- The Brenner weight is simply the standard weight divided by the respective observed weight. If the observed weight was zero, we use a zero Brenner weight to avoid dividing by zero. E.g.

SEX	ag_icss	observed_weight	standard_weight	brenner_weight
1	15 - 19	0.0035194	0.00164	0.4659937
1	20 - 24	0.0048839	0.00215	0.4402239
1	25 - 29	0.0068548	0.00416	0.6068696
1	30 - 34	0.0088307	0.00842	0.9534962
1	35 - 39	0.0123936	0.01874	1.5120746
1	40 - 44	0.0190597	0.03489	1.8305632
1	45 - 49	0.0326613	0.04906	1.5020815
1	50 - 54	0.0581662	0.07094	1.2196082
1	55 - 59	0.0973905	0.09641	0.9899325
1	60 - 64	0.1399647	0.13359	0.9544547
1	65 - 69	0.1685793	0.14234	0.8443504
1	70 - 74	0.1703019	0.14766	0.8670485
1	75 - 79	0.1369002	0.14131	1.0322116
1	80 - 84	0.0841388	0.09347	1.1109019
1	85 - Inf	0.0488469	0.05522	1.1304720
2	15 - 19	0.0034047	0.00164	0.4816912
2	20 - 24	0.0050961	0.00215	0.4218913
2	25 - 29	0.0082776	0.00416	0.5025614
2	30 - 34	0.0140885	0.00842	0.5976495
2	35 - 39	0.0234333	0.01874	0.7997156
2	40 - 44	0.0383424	0.03489	0.9099587
2	45 - 49	0.0608126	0.04906	0.8067413
2	50 - 54	0.0799317	0.07094	0.8875073
2	55 - 59	0.0962285	0.09641	1.0018860
2	60 - 64	0.1175660	0.13359	1.1362977
2	65 - 69	0.1280586	0.14234	1.1115225
2	70 - 74	0.1298138	0.14766	1.1374756
2	75 - 79	0.1189789	0.14131	1.1876896
2	80 - 84	0.0923624	0.09347	1.0119919
2	85 - Inf	0.0771244	0.05522	0.7159858

- Although the weights could be assigned to each individual cancer record, we apply the weights only at the estimation stage. This is still not direct adjusting — we don't estimate first by age group and then compute a weighted average. Instead we compute summary statistics (such as time at risk `t_at_risk`) by age group, apply Brenner weights to those, sum them together over the age groups, and compute the survival estimates from them. This is mathematically identical to assigning weights to individual records but easier to understand and debug (and maybe marginally faster). For more details on the survival estimation itself, see section **Survival estimation**.

Tables of survival estimates

- All tables are stratified by sex, cancer entity, and follow-up time.
- Survival estimation is based on monthly intervals up to 5 years of follow-up unless specifically declared otherwise.
- Unless specificed otherwise, age at diagnosis must be at least 15. If a person was diagnosed on their 15th birthday (happy birthday), they are included.

Possible columns in each table of survival estimates:

- `n_in_follow_up_at_interval_start`: Number of records in follow-up at the start of the interval.
- `n_in_follow_up_at_interval_stop`: Number of records in follow-up at the end of the interval.
- `t_at_risk`: Total amount of time at risk in the interval.
- `n_events`: Total number of events in the interval.
- `n_events_exp_e2`: Total number of expected events in the interval according to the Ederer II method.
- `NS_pp_ch`: Pohar Perme estimate at the end of the interval.
- `S_ch`: Observed survival estimate at the end of the interval.
- `*_se`: Standard error of the corresponding estimate column. E.g. `NS_pp_ch_bw_se`.
- `*_lo`: Lower confidence bound of the corresponding estimate column.
- `*_hi`: Upper confidence bound of the corresponding estimate column.
- `*_bw`: Brenner weighted, i.e. age-adjusting has been applied. E.g. `NS_pp_ch_bw`.

`survival_by_POI`

- Additional stratifying column `POI`, period of incidence.
- Contains age-adjusted Pohar Perme estimates.

`survival_by_POI_et_ECIS_AG_SURV`

- Additional stratifying column `POI` and `ECIS_AG_SURV`.
- Contains age-adjusted Pohar Perme estimates *not* adjusted by age.

`survival_latest_period`

- Period analysis estimates for the latest period.
- Contains age-adjusted Pohar Perme estimates.
- This table is always produced, but all the estimates are missing (NA) unless there are at least two contributing combinations of year of incidence (YoI) and calendar year of follow-up (CYoF) for each year of survival time (YoST). The table below is an example for period analysis of 2021-2024.

YoI	CYoF=2021	CYoF=2022	CYoF=2023	CYoF=2024
2017	[4.0, 5.0[			
2018	[3.0, 4.0[	[4.0, 5.0[		
2019	[2.0, 3.0[	[3.0, 4.0[	[4.0, 5.0[	
2020	[1.0, 2.0[	[2.0, 3.0[	[3.0, 4.0[	[4.0, 5.0[
2021	[0.0, 1.0[	[1.0, 2.0[	[2.0, 3.0[	[3.0, 4.0[
2022		[0.0, 1.0[	[1.0, 2.0[	[2.0, 3.0[
2023			[0.0, 1.0[	[1.0, 2.0[
2024				[0.0, 1.0[

- This criterion is tested for as follows:
  - Create a table which displays every combination of `YoI`, `CYoF`, and `YoST`. `YoI` values are based on `crate_user_settings` values for `incidence_first_year` and `incidence_last_year`. `CYoF` values are those that define the period itself, e.g. 2021-2024, except the upper limit is at most the

crate_user_settings value for survival_follow_up_last_year. YoST values are derived from the subtraction CYoF - YoI.
  - We count the number of appearances of each YoST value, e.g. [0.0, 1.0[might appear three times etc.
  - We allow for this estimation if every YoST value from [0.0, 1.0[to [4.0, 5.0[appears at least twice and otherwise not.
  - From the example table above you can see that period analysis is not conducted if e.g. one has YoI values 2017-2021 and CYoF values 2021-2022 because then the first YoST only has one contributing combination.

survival_latest_period_by_ECIS_AG_SURV

- Otherwise the same as survival_latest_period, but additionally stratified by age group column ECIS_AG_SURV.
- The Pohar Perme estimates are *not* adjusted for age.

survival_by_ECIS_POI

- The purpose of this table is to have estimates as close to the ones produced in CancerCare some time ago so we can validate our approach against some known estimates.
- Additional stratifying column ECIS_POI which has one value, 2000-2007.
- Follow-up is additionally limited to the years 2000-2009 identically to what was done in CancerCare.
- Contains age-adjusted Pohar Perme estimates.

Childhood survival

Tables of childhood survival estimates

`survival_childhood_latest`

- All tables are stratified by sex, ICCC cancer entity, and follow-up time.
- Period analysis estimates for the latest period allowed in CRATE.
- Limited to cases where the patient was < 15 years old at the time of diagnosis.
- Contains (unadjusted) observed survival estimates.
- This table is always produced, but all the estimates are missing (NA) unless there are at least two contributing combinations of year of incidence (YoI) and calendar year of follow-up (CYoF) for each year of survival time (YoST). The table below is an example for period analysis of 2021-2024.

YoI	CYoF=2021	CYoF=2022	CYoF=2023	CYoF=2024
2017	[4.0, 5.0[			
2018	[3.0, 4.0[	[4.0, 5.0[		
2019	[2.0, 3.0[	[3.0, 4.0[	[4.0, 5.0[	
2020	[1.0, 2.0[	[2.0, 3.0[	[3.0, 4.0[	[4.0, 5.0[
2021	[0.0, 1.0[	[1.0, 2.0[	[2.0, 3.0[	[3.0, 4.0[
2022		[0.0, 1.0[	[1.0, 2.0[	[2.0, 3.0[
2023			[0.0, 1.0[	[1.0, 2.0[
2024				[0.0, 1.0[

- This criterion is tested for as follows:
  - Create a table which displays every combination of `YoI`, `CYoF`, and `YoST`. `YoI` values are based on `crate_user_settings` values for `incidence_first_year` and `incidence_last_year`. `CYoF` values are those that define the period itself, e.g. 2021-2024, except the upper limit is at most the `crate_user_settings` value for `survival_follow_up_last_year`. `YoST` values are derived from the subtraction `CYoF - YoI`.
  - We count the number of appearances of each `YoST` value, e.g. `[0.0, 1.0[` might appear three times etc.
  - We allow for this estimation if every `YoST` value from `[0.0, 1.0[` to `[4.0, 5.0[` appears at least twice and otherwise not.
  - From the example table above you can see that period analysis is not conducted if e.g. one has `YoI` values 2017-2021 and `CYoF` values 2021-2022 because then the first `YoST` only has one contributing combination.

`survival_childhood_latest_by_AG_SURV_CHILDHOOD`

- Like `survival_childhood_latest` but additionally stratified by column `AG_SURV_CHILDHOOD` which has levels `0-4`, `5-9`, and `10-14`.

Prevalence

Prevalence estimation

The following steps are performed:

- Only keep records that meet all of the following:
  - follow-up time is > 0
  - `FLAG_SURV` contains no `ERR-X` codes
  - `SEX` is 1 or 2
  - `BOD != 0`
  - `AUTOPSY != 1`
  - `incidence_first_year <= YOI <= incidence_last_year` from `crate_user_settings`
  - `YOI <= prevalence_observation_year` from `crate_global_settings`.
- For each cancer entity separately, take the first cancer record for each patient of the current entity. This is performed *after* the filtering above. E.g. if a patient has three diagnoses for an entity but the first one was filtered out, then we only use the second diagnosis for that patient.
- Estimate prevalence by calling `popEpi::prev_lexis`. For the purpose of computing the "effective prevalence" which includes also the expected number of prevalent patients of those who were censored, we compute observed survival estimates on-the-fly.
- The following specifications are used for producing the table of observed survival estimates:
  - Period analysis is used. The period is defined as the ten years before the last observation time point in the analysis. E.g. for observation time point 2020-21-31 we use the period 2011-2020.
  - Follow-up time is initially split into
    - monthly intervals up to 5 years of follow-up,
    - annual intervals up to at most 100 years of follow-up (which I'm sure is never reached).
  - With a table of aggregated statistics in hand, the age groups are re-aggregated as needed to ensure that every survival curve has more than 100 years of `t_at_risk` in total. This is particularly likely to happen when we want to produce prevalence estimates in 5-year age groups. E.g. 20 patients with 5 years would do that. E.g. if the age group 0-4 has only 10 years and 5-9 has 300 years then a new age group 0-9 is formed with 310 years.
  - The follow-up time intervals are re-aggregated to ensure `n_events > 0` is true for every interval. E.g. if the second interval has no events then it may be combined with the third one, which results in one longer interval based on the two. This interval aggregation has no limit, so a dataset with zero patients results in one very wide survival interval.
  - Observed survival is estimated by estimating the piecewise constant hazard by survival interval and then deriving the survival estimates from those.
  - Because this table of survival estimates is used for the extrapolation, we want to make sure there are estimates available at reasonable intervals even if we had to estimate survival using rather wide ones. To this end we interpolate the survival estimates, so that each survival curve will in the end have 1000 interval regardless of how many were used in the estimation. This interpolation is linear, e.g. if we have estimated `S_ch = 0.6` at `T = 2` and `S_ch = 0.5` at `T = 3`, then at `T = 2.5` we have the interpolated survival `S_ch = 0.55`.
- The confidence intervals for prevalence proportions are based on the log transform.

Tables of prevalence estimates

- All tables are stratified by sex, cancer entity, observation time point, and time from diagnosis.

Possible columns in each table of prevalence estimates:

- `n_qi_5`: The number of patients whose follow-up was censored before the prevalence observation time point. This is the number of patients for which we computed the expected number still at risk at the prevalence observation time point.
- `nd_qi_5`: `n_qi_5` + `n_prev`.
- `p_qi_5`: `n_qi_5 / nd_qi_5`.
- `n_at_risk`: The size of the general population at the prevalence observation time point.
- `n_prev`: Number of prevalent patients at the prevalence observation time point.
- `p_prev`: Proportion of prevalent patients relative to the population size.
- `n_prev_eff`: Effective number of prevalent patients. This is `n_prev` + the number of "extrapolated" patients at the prevalence observation time point.
- `p_prev_eff`: As `p_prev`, but based on `n_prev_eff`.
- `*_europe_2013`: Implies that the estimates have been adjusted for age using direct adjusting and the "europe_2013" standard weights. E.g. `p_prev_eff_europe_2013`.
- `*_europe_1976`: Implies that the estimates have been adjusted for age using direct adjusting and the "europe_1976" standard weights. E.g. `p_prev_eff_europe_1976`.
- `*_se`: Standard error of the corresponding estimate column. E.g. `p_prev_eff_europe_1976_se`.
- `*_lo`: Confidence interval lower bound of the corresponding estimate column. E.g. `p_prev_eff_europe_1976_lo`.
- `*_hi`: Confidence interval upper bound of the corresponding estimate column. E.g. `p_prev_eff_europe_1976_hi`.

`prevalence_latest`

- This table simply contains the prevalence estimates for the last allowed date. E.g. for 2020-12-31.

`prevalence_latest_by_ts_age`

- As `prevalence_latest`, but additionally stratified by age group (based on age at prevalence observation time point). The estimates are not adjusted for age.

Technical information about this document

- This document was created at 2026-10-29 16:44:50.897797.
- The latest commit was b598c4e23a4e1f4b4db57b254055918cff4d2df8.