Real-Time Surveillance of Infectious Diseases: A Statistical Perspective

Richard Aubrey White

Contents

Chapter 1

Real-Time Surveillance of Infectious Diseases: A Statistical Perspective

Chapter 2

Naming conventions

2.1 Short summary

- File name:
  - location: use location_code for data and location_name_nb_utf for results (report, figures)
  - time generated: add yyyy_mm_dd and time in a day, if needed.
- Task name: name_grouping + name_action + name_variant
- Database (DB) table name: name_access + name_grouping + name_variant
- DB table variable name
  - unified variables
  - task-specific variables

The naming should all follow the rules outlined below.

2.2 General rules

2.2.1 Language

English is the language for our code. We use **English** in Sykdomspulsen Analytics.

Names that are abbreviations or in Norwegian are kept as they are: data sources such as msis, dar, sysvak, normomo, locations such as viken, oslo.

For example:

- Task names: oh_import_msis_data, covid19_risk_level_export_maps_msis_cases_county

- Database table names: `anon_covid19_risk_levels`

Whether to use location name, see the section on *file name*.

2.2.2 Ordering of variables

Sometimes variables need to be ordered. Variables should be ordered as follows:

- time
- location
- age
- sex

e.g. A database table could be called `msis_by_time_location_age_sex` or a filename could be called `2020_oslo_05-10_male.xlsx`

2.2.3 Ages

Ages should always contain at least 2 digits and are joined by a hyphen (e.g. `05-10`).

Use `85+` instead of `>=85`.

These tips will help your data be easily sorted and kept in the right order.

2.2.4 Capital letters

Capital letters are to be avoided whenever possible. This is also the case in filenames (e.g. `data.rds` is preferred to `data.RDS`)

2.2.5 snake_case or camelCase?

Use snake_case.

2.3 File name

2.3.1 `location_code` or `location_name`?

- Data files: use `location_code`

When saving data files for use within Sykdomspulsen Analytics, `location_code` (e.g. `county03`) should be used, such as `county03_05-10_male_2020.rds`.

- Results: use `location_name`

This applies to figures, tables in excel sheets, documents. Use the actual location name because they will be read by people outside the Sykdomspulsen team.

Use `location_name_nb_utf` for the Norwegian Bokmål language and UTF-8 encoding. For instance `2020_Oslo_fylke_05-10_male.xlsx`.

You can get `location_name_nb_utf` from the function `fhidata::norway_locations_names()`

2.3.2 When is the file created

In results (e.g. reports), an indicator of time when the files are created are necessary. It allows us to find which one is the most recent version of files with the same names, and it allows easy tracking of an Airflow error.

e.g. `Epidemiologisk_situasjonsrapport_2021_05_31_0659.docx` for a report generated on May 31, 2021 06:59 AM.

2.4 DB table variables: unified variables

All accessible DB tables will contain these 16 variables.

Time conversion functions can be found in package fhiplot.

Variables

Accepted values

Description

granularity_time

total, isoyear, isoweek, day, hour, calyear, calmonth

Temporal granularity

granularity_geo

norge, county, municip, baregion, wardoslo, wardbergen, wardtrondheim, wardstavanger, missingwardoslo, missingwardbergen, missingwardtrondheim, missingwardstavanger, region, faregion, notmainlandmunicip, notmainlandcounty, missingmunicip, missingcounty

Geographical granularity

country_iso3

nor, den, swe, fin

ISO3 country code, e.g. nor for Norway

location_code

norge, countyXX, municipXXXX, . . .

The geographical location

border

2020

The borders (kommunesammenslåing) that `location_code` represents age

e.g. 0, 1, 2, 00-04, 05-14, total

Age in years

sex

male, female, total

Sex

isoyear

YYYY

Use function `fhiplot::isoyear_n`

isoweek

0, 1, 2, . . . , 53

Use function `fhiplot::isoweek_n`

isoyearweek

YYYY-WW

Use function `fhiplot::isoyearweek_c`

season

e.g. 2014/2015, 2014/2014, 2014

Winter seasons are denoted by 2014/2015 (generally three options: weeks 30-29, weeks 40-20, or weeks 40-39), summer seasons are denoted by 2014/2014 (generally weeks 21-39), years are denoted by the year.

seasonweek

1, 2, 3, . . . , 23, 23.5, 24, . . . , 52

Use function `fhiplot::seasonweek_to_week_n()`

calyear

e.g. 2021

Calendar year. Cal* == NA when granularity_time == iso*; cal* != NA when granularity == day or cal*

calmonth

1, . . . , 12

Calendar month.

calyearmonth

e.g. 2021-03

Calendar year month.

date

YYYY-MM-DD

Always corresponds to the last date in the time period. E.g. if `granularity_time=='week'` then date is the Sunday of that week. `fhidata::days` is useful here.

2.5 DB table variables: task-specific variables

Task-specific variables are commonly referred to as Tags. A `tag` is a descriptor of the row. That is, an identifier of data that is in long format. For example: `tag_outcome, tag_exposure`.

Variable names that contain values should be in one the following forms:

- `descriptive_format`
- `descriptive_format_functiontime`

Tag names

Examples

Notes

descriptive

deaths, consultations

msis_case_testdate

modelled_r_greater_than_1

format

n, cum_n

counts, cumulative counts

pr100, pr100000

percentages/proportions

n_expected, pr100_expected

corrected_n, corrected_baseline_n

predinterval_p02x5, confinterval_p00x1

p02x5, p97x5 (0.025, 0.975)

predinterval_l2, confinterval_u2

lower and upper limits (see explanation below) status

functiontime

sum0_13

aggregated n case for today and preceeding 13 days

2.5.0.1 Tag descriptive

When creating a database table with many different sources of data, it is recommended to include the data source as a descriptive:

- `msis_cases_testdate_n` - number of cases by testing date
- `msis_cases_regdate_n` - number of cases by registration date
- `msislab_tests_n` - number of tests performed
- `nipar_hospital_main_cause_n` - number of people who were admitted to the hospital with covid-19 as a main cause
- `nipar_icu_n` - number of people who were admitted to the ICU with covid-19
- `mor_deaths_n` - number of deaths

2.5.0.2 Tag format

`format` is the format that the value is stored in.

For **percentages/proportions**, we recommend using `pr100` (i.e. values going between 0 and 100) over `pr1` (i.e. values going between 0 and 1).

For **intervals**, use `XintervalY`. We recommend:

- `X` describing the type of interval. If you are using frequentist statistics, this is probably `predinterval` or `confinterval`. If you are using Bayesian statistics, then `credinterval` would be appropriate, and the `descriptive` would then determine if the credible interval is describing a variable (Bayesian prediction interval) or a population/distribution parameter (Bayesian confidence interval).
- `Y` describes the limit, for example (in percentiles) `_p02x5`, `_p97x5`, `_p00x1`, `_p99x9` or (in z-scores) `_l2`, `_u2`, `_l4`, `_u4`.

Some examples are as follows:

- `deaths_n` - number of deaths
- `deaths_n_sum0_13` - number of deaths that occurred today and the preceeding 13 days
- `deaths_pr100000_sum0_13` - number of deaths that occurred today and the preceeding 13 days, per 100 00 population
- `consultations_r80_vk_ote_n` - number of influenza consultations
- `deaths_registered_n` - number of deaths registered
- `deaths_corrected_n` - number of deaths (corrected for registration delay)

- `deaths_corrected_n_credinterval_p02x5` - Bayesian prediction interval for the “true” number of deaths
- `deaths_corrected_n_credinterval_p97x5` - Bayesian prediction interval for the “true” number of deaths
- `deaths_corrected_baseline_n` - Expected number of deaths (i.e. the baseline)
- `deaths_corrected_baseline_n_credinterval_p02x5` - Bayesian prediction interval for the expected number of death (baseline)
- `deaths_corrected_baseline_n_credinterval_p97x5` - Bayesian prediction interval for the expected number of death (baseline)

2.5.0.3 Tag functiontime

We can also include variables that refer to multiple days/weeks:

- `msis_cases_testdate_n_sum0_13 -` the aggregate number of cases for today and the preceeding 13 days by testing date
- `msis_cases_testdate_pr100000_sum0_13 -` the aggregate number of cases for today and the preceeding 13 days by testing date per 100 000 population