Consortium for
Statistics in
Disease Surveillance

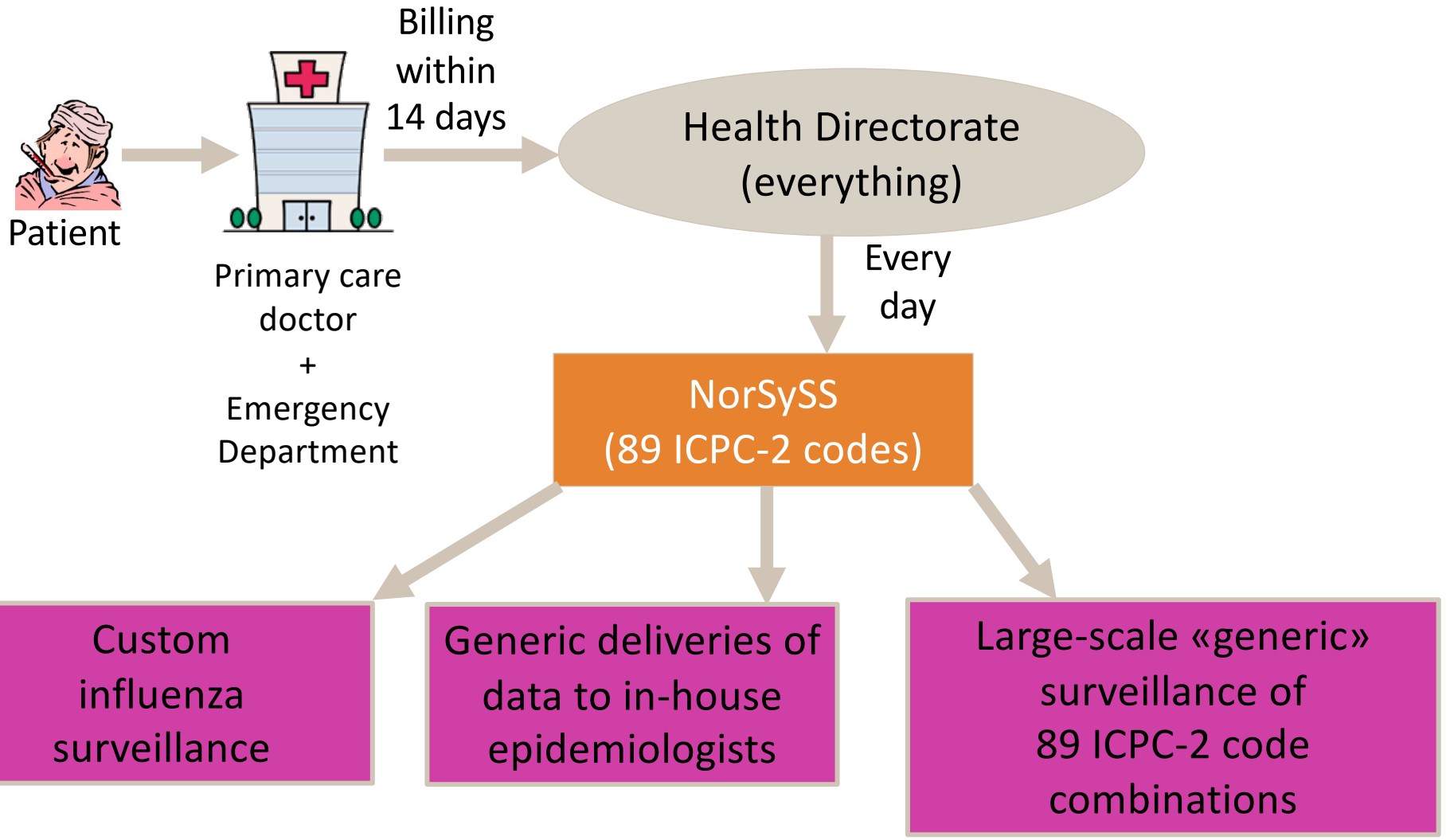
NIPH
Norwegian Institute of Public Health

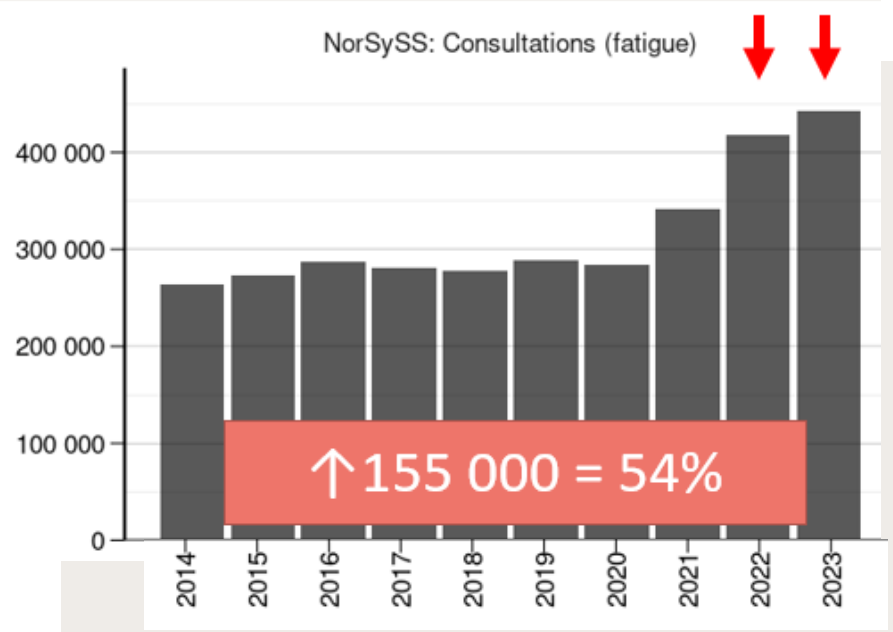
Core Surveillance 9 as a framework for real-time analysis and disease surveillance, using NorSySS as an example

Richard Aubrey White, Ph.D.

Northern European Symposium on Automated Surveillance, 2023-12-07

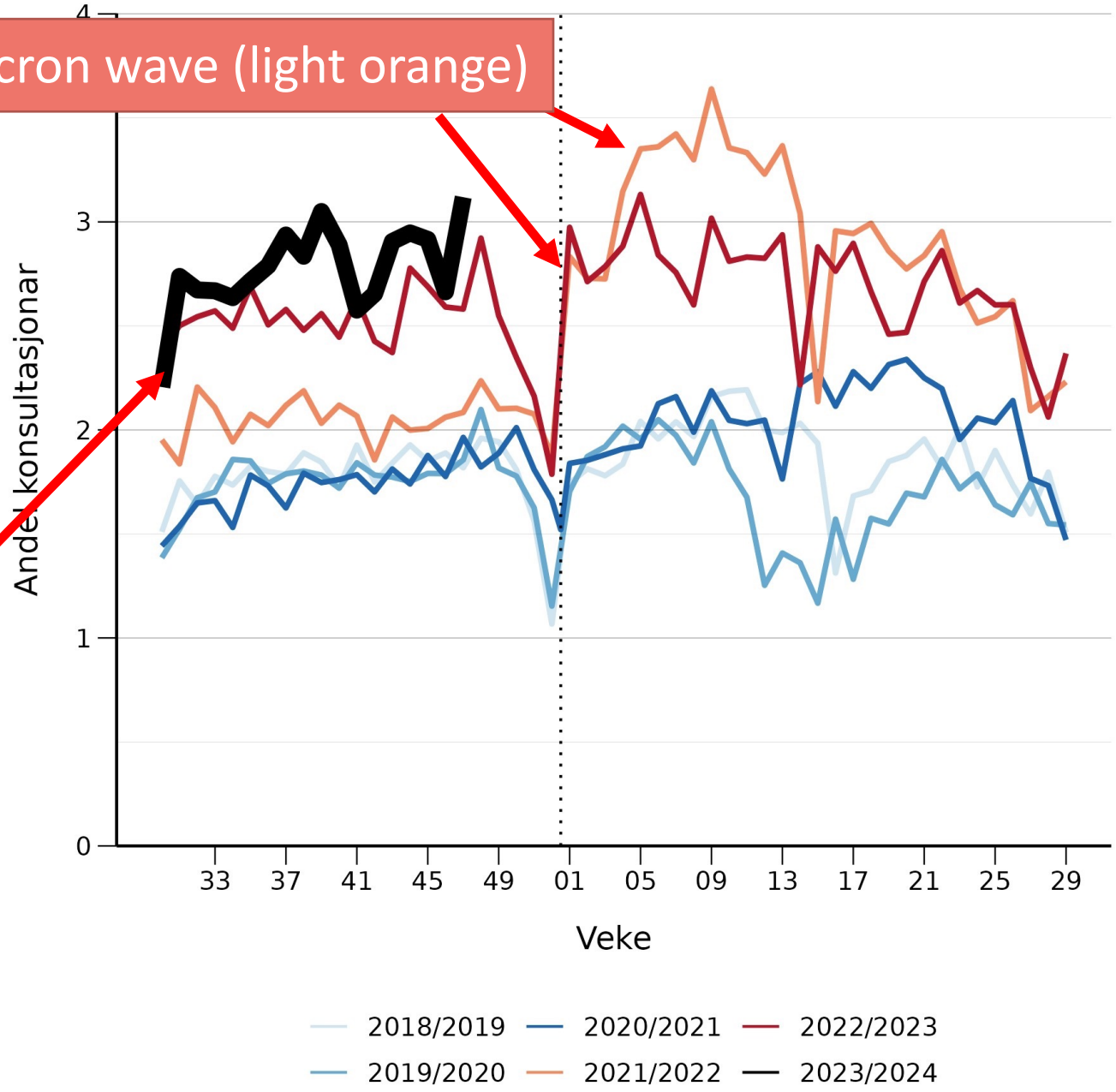
Norwegian Syndromic Surveillance System (NorSySS) in brief





First omicron wave (light orange)

Still high (black line)



Number/percentage of primary healthcare consultations that are for fatigue

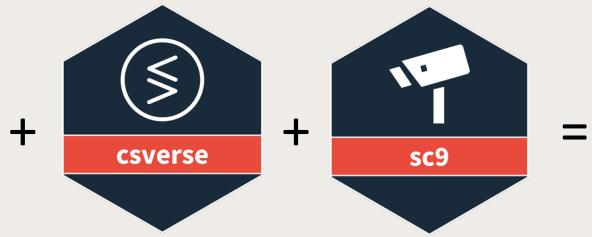
Then -> Now

	2022	2023
Employees	8	2
Data sources	15 (including NorSySS)	1 (NorSySS)
Number of ICPC-2 codes analysed in NorSySS	10	89
Analyses performed	A lot of custom analyses	Focus on generic/repeatable surveillance
Infrastructure	Bundled together	- CSIDS provides generic infrastructure - NorSySS uses CSIDS' generic infrastructure

Publicly available open-source infrastructure + framework

Consortium for Statistics in Disease Surveillance

Linux + R + R-packages + quarto + ...



3 GB

csids/cs9-su-csverse

docker

Closed-source implementation of NorSySS

Apache Airflow

Shared database

localhost/interactive

localhost/workbench

GitHub

norsyss.cs9
interactive branch
(implementation)

+

csids/cs9-su-csverse

RStudio®

+

localhost/interactive

Apache Airflow

localhost/auto

GitHub

norsyss.cs9
auto branch
(implementation)

+

csids/cs9-su-csverse

Generic surveillance

89 ICPC-2 code combinations

```
> norsyss::icpc2[1:5]
   icpc2raw_tag icpc2group_tag  icpc2group_description_en        component_en           bodysystem_en
1:          A02            a02                     Chills Symptoms/Complaints General and unspecified
2:          A03            a03                      Fever Symptoms/Complaints General and unspecified
3:          A04            a04 Weakness/tiredness general Symptoms/Complaints General and unspecified
4:          A05            a05                Feeling ill Symptoms/Complaints General and unspecified
5:          A71            a71                    Measles          Infections General and unspecified
   icpc2group_description_nb component_nb           bodysystem_nb icpc2group_description_nn component_nn
1:                Frysninger    Symptomer Allment og uspesifisert                Frysningar      Symptom
2:                     Feber    Symptomer Allment og uspesifisert                     Feber      Symptom
3:         Slapphet/tretthet    Symptomer Allment og uspesifisert      Slappheit/trøyttleik      Symptom
4:            Sykdomsfølelse    Symptomer Allment og uspesifisert             Sykdomsfølelse      Symptom
5:                 Meslinger  Infeksjoner Allment og uspesifisert                  Meslingar  Infeksjonar
            bodysystem_nn includes_influenza_covid19 has_historical_data
1: Allment og uspesifisert                    FALSE                TRUE
2: Allment og uspesifisert                    FALSE                TRUE
3: Allment og uspesifisert                    FALSE                TRUE
4: Allment og uspesifisert                    FALSE                TRUE
5: Allment og uspesifisert                    FALSE                TRUE
```

```
> norsyss::reports
$descriptions
                report_tag                                    report_name_en
1:                selected                                  Selected diseases
2: general_and_unspecified                               General and unspecified
3:        blood_and_immune Blood, blood forming organs, and immune mechanism
4:               digestive                                         Digestive
5:                     eye                                               Eye
6:                     ear                                               Ear
7:             respiratory                                       Respiratory
8:                    skin                                              Skin
9:                     all                                               All

$icpc2
$icpc2$selected
[1] "a78"                   "a04"                   "gastroenteritis"       "resp
[5] "covid19"               "r80"                   "r72"

$icpc2$general_and_unspecified
 [1] "a02" "a03" "a04" "a05" "a71" "a72" "a73" "a74" "a75" "a76" "a77" "a78"

$icpc2$blood_and_immune
[1] "b02" "b70" "b71"

$icpc2$digestive
 [1] "gastroenteritis" "d01"             "d02"             "d06"             "d08"
 [6] "d09"             "d10"             "d11"             "d14"             "d18"
[11] "d25"             "d29"             "d70"             "d73"             "d87"
[16] "d99"

$icpc2$eye
[1] "f70" "f73"
```

Scale of NorSySS' "generic surveillance"

- Cleaned data: 1 TB / 1,4 billion rows of data (x2)
  - 373 locations
  - 9 age groups
  - 89 ICPC-2 codes / 6 consultation types
  - 18 years
  - 20 CPU cores = 14 hours to clean
- Results for 24 million analyses (x2)
  - 16 locations
  - 9 age groups
  - 89 ICPC-2 codes / 1 consultation type
  - 936 weeks
  - 12 million short term trends with 20 CPU cores = 2h 20 min
  - 12 million signal detections with 20 CPU cores = 14 min
- 30 000 figures created (x2)
  - 16 locations
  - 89 ICPC-2 codes / 1 consultation type
  - 11 figure variants
  - 2 file locations
  - 20 CPU cores = 40 minutes

Weekly report(s)

- 9 weekly reports
- Divided on disease groups
- Millions of analyses
  - By disease
  - By age
  - By location
  - By time
- All automatically generated
- Created every day
- Emailed 1x/week

INTERACTIVE: Vekerapport 2023-47 frå NorSySS

N

NorSySS <norsyss@fhi.no>
To NorSySS

vekerapport_frå_norsyss_utvalde_sjukdommar.pdf
6 MB

Velkommen til vekerapporten frå NorSySS.

Vedlagt finn du rapporten som dekker utvalde sjukdommar, og under finn du lenkjer til ra

Ver obs på at for å klikka på rapportlenkjene må du ha tilgang til N:/norsyss_vekerapport.

- Utvalde sjukdommar (7 diagnosekodar)
- Allment og uspesifisert (12 diagnosekodar)
- Blod, bloddannande organ og immunsystemet (3 diagnosekodar)
- Fordøyingssystemet (16 diagnosekodar)
- Auge (2 diagnosekodar)
- Øre (6 diagnosekodar)
- Luftvegar (34 diagnosekodar)
- Hud (16 diagnosekodar)
- Alle (89 diagnosekodar)

Mvh
NorSySS-teamet

Weekly report(s)

52 analyses

144 analyses

104 analyses

234 analyses

832 analyses

0.6 Covi

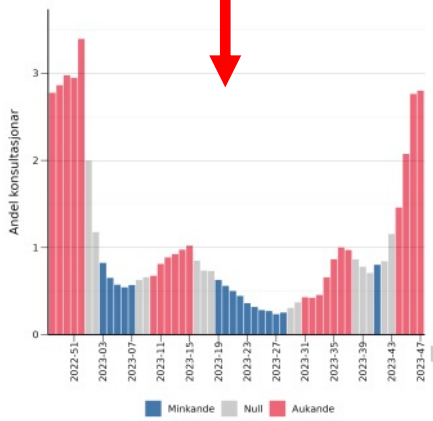
0.6.1 ICP

R991, R992

ℹ Historiske data

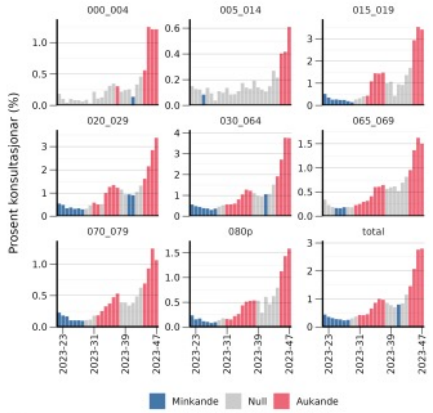
Desse ICPC-2-kodane h ikkje nok historiske data for å laga baseliner

0.6.2 Trendar og nivå på landsbasis



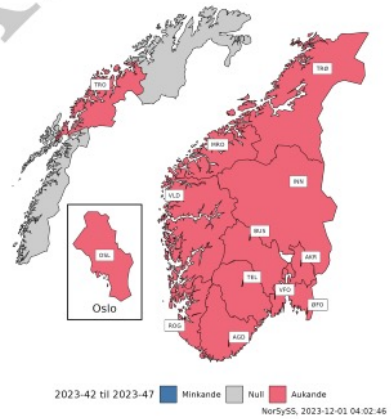
Figur 17: Covid-19 — Epikurve farga etter seks vekers trend for andel konsultasjonar på nasjonalt nivå.

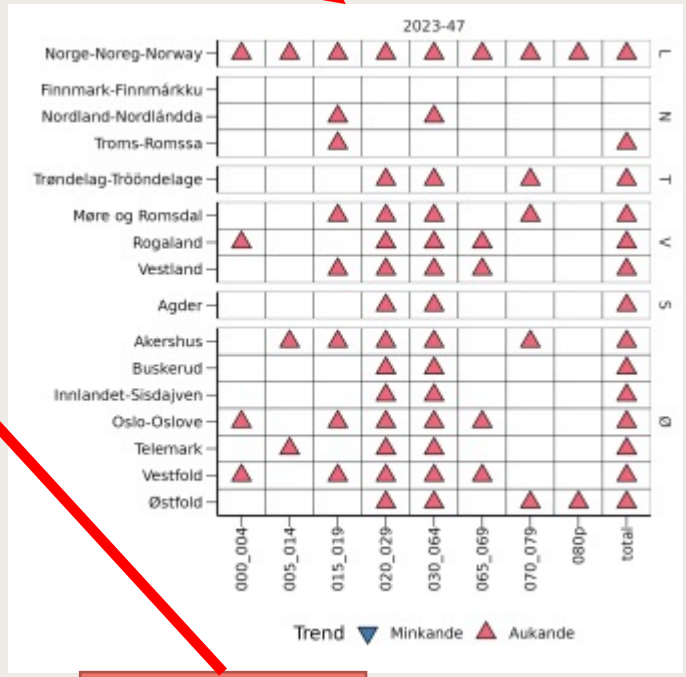
0.6.3 Trendar og nivå etter tid og aldersgrupper

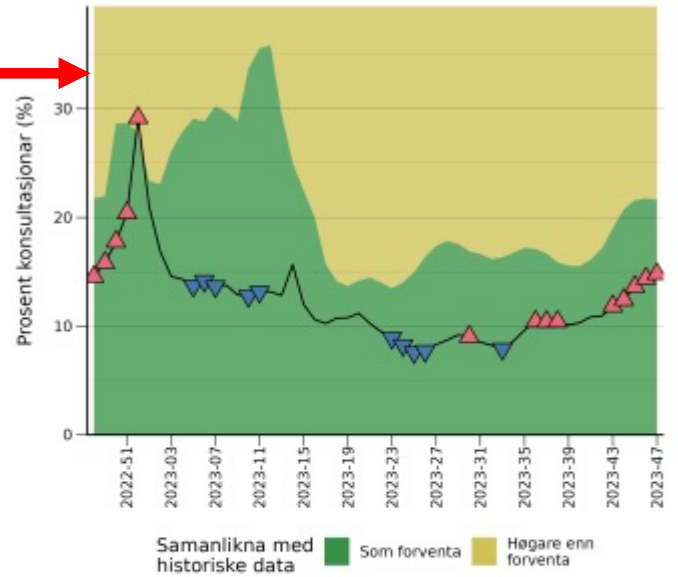


Figur 18: Covid-19 — Epikurve farga etter seks vekers trend for andel konsultasjonar på nasjonalt nivå.

0.6.4 Trendar og nivå etter tid og fylke





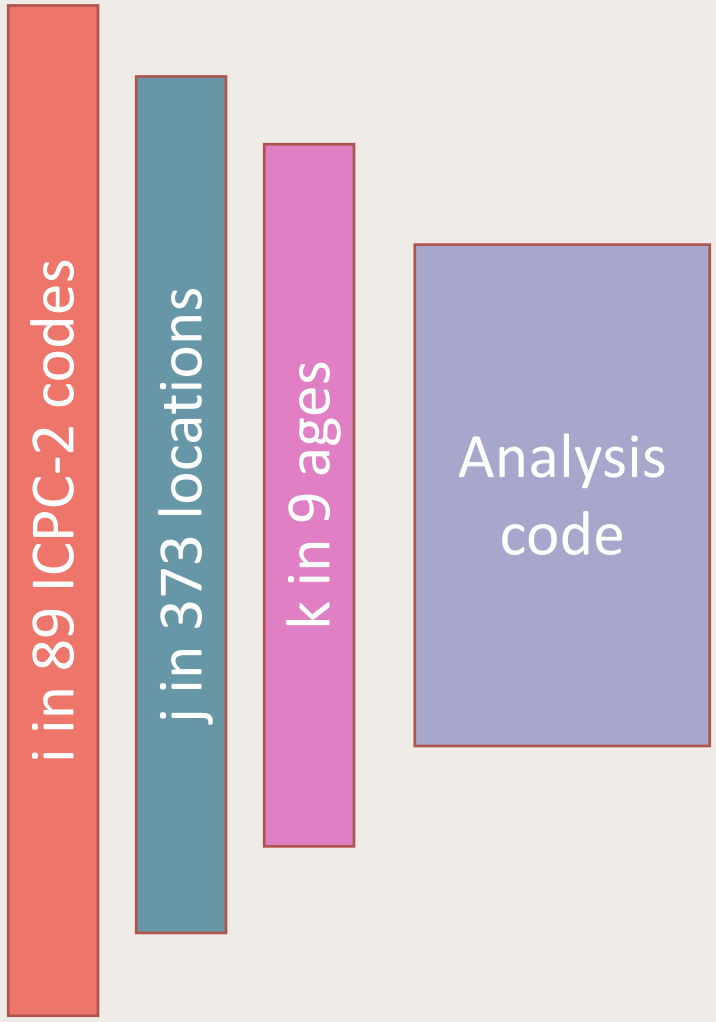


How is this possible?

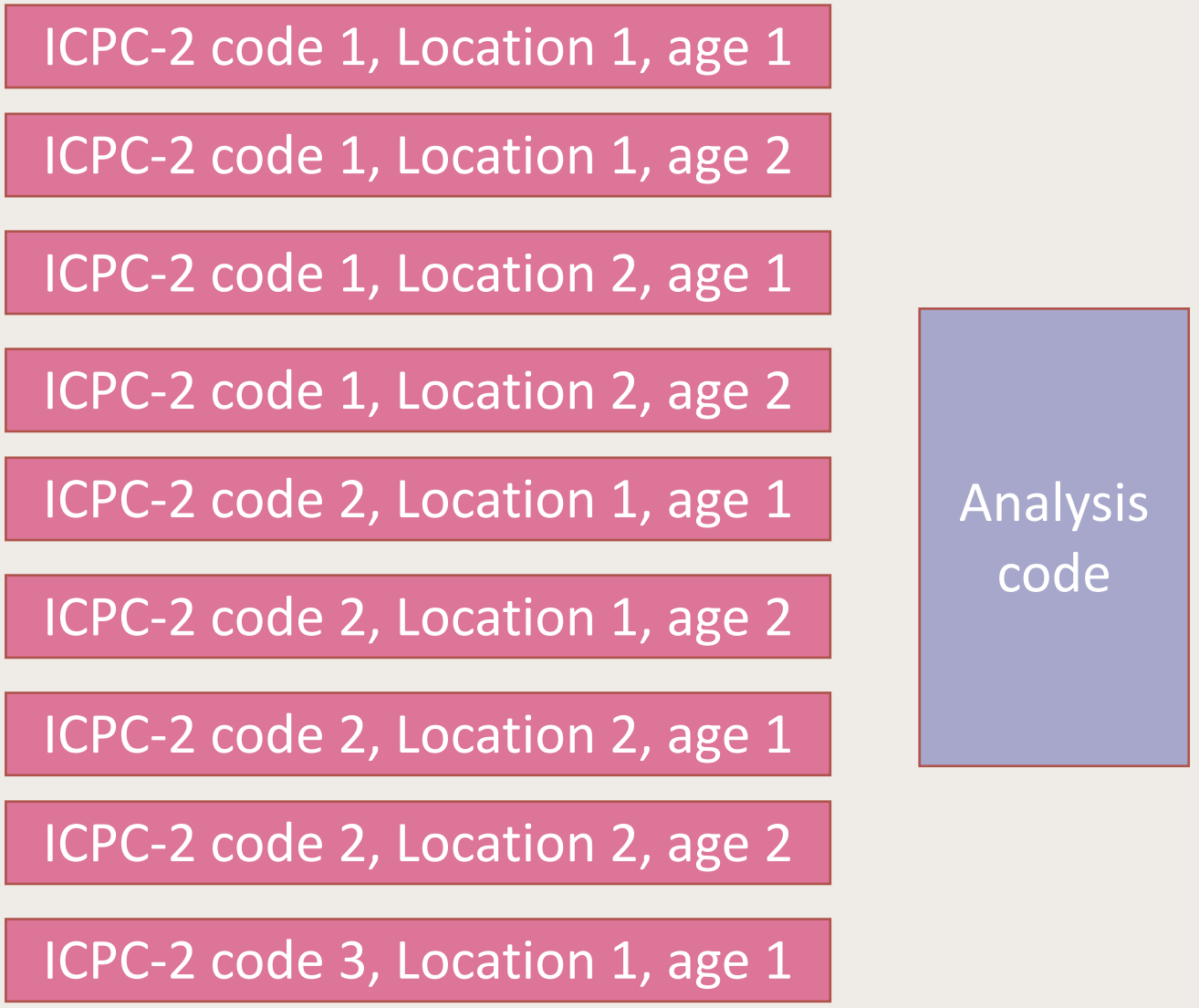
Core Surveillance 9

- A free and open-source framework for real-time analysis and disease surveillance
- Under development for 9 years
- Can handle millions of complex statistical analyses and terabytes of data
- Good for both scheduled analyses and ad-hoc analyses
- Designed to work exactly like a single interactive independent script on your computer (line 1, line 2, line 3…)
- No loops, minimal amount of data, the “core of the problem”
  - “I have been given data for 0-4 year old males in Oslo in 2012, what analysis will I run on this data?”
- Does not need high R levels to work on editing the task

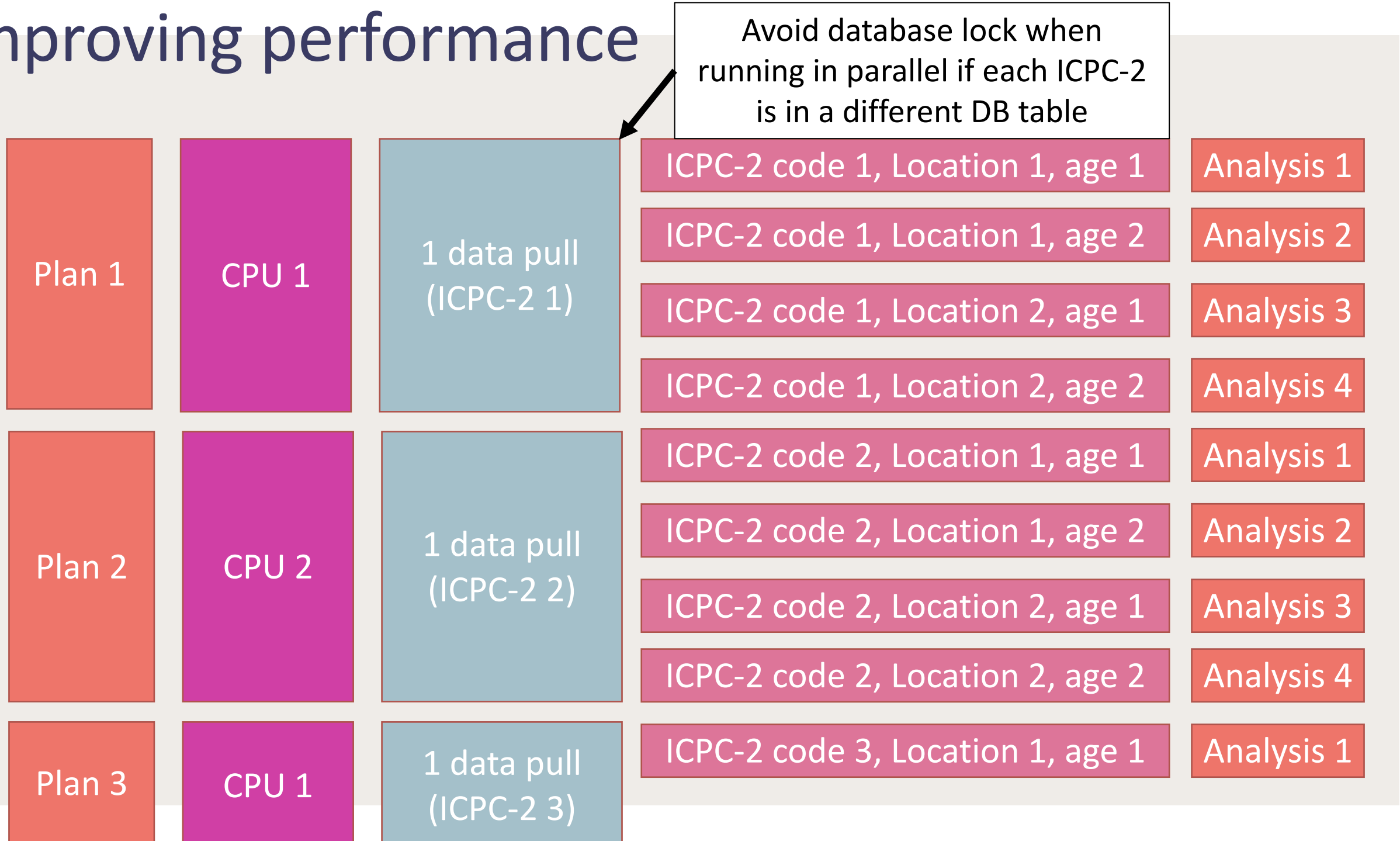
Multiple loops



One loop

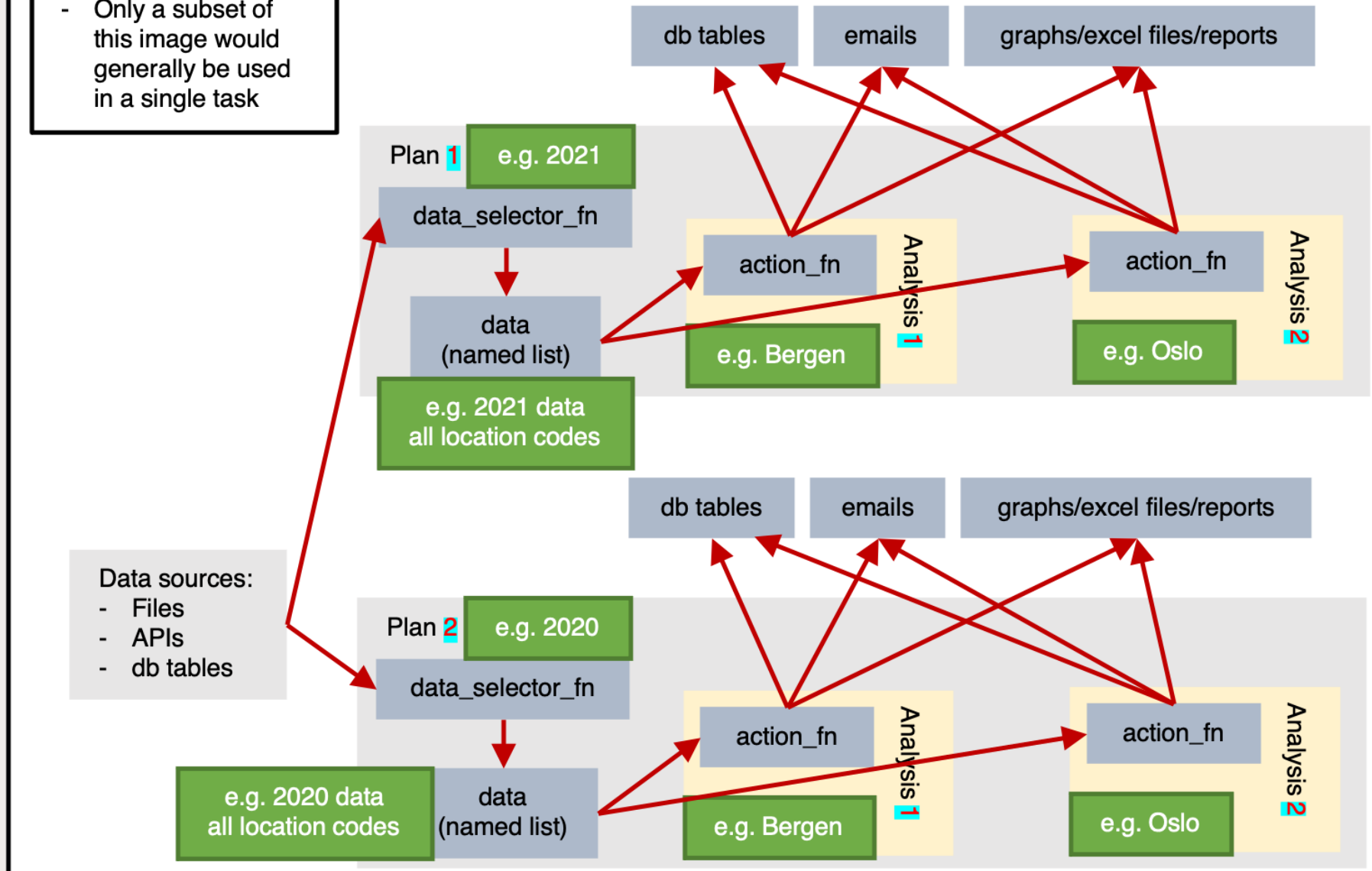


Improving performance



What is a “task” in Surveillance Core 9?

General task
- Shows all possibilities
- Only a subset of this image would generally be used in a single task



Partitioned tables avoid database lock

Core Surveillance 9

Adding a new database table

Easily define column names/types

Columns that uniquely identify rows

Indexes (for speed)

Automatic data validation

```
## > anon_norsyss_data ----
global$ss$add_partitionedtable(
  name_access = c("anon"),
  name_grouping = "norsyss",
  name_variant = "data",
  name_partitions = norsyss::icpc2$icpc2group_tag,
  column_name_partition = "icpc2group_tag",
  field_types =  c(
    "granularity_time" = "TEXT",
    "granularity_geo" = "TEXT",
    "country_iso3" = "TEXT",
    "location_code" = "TEXT",
    "border" = "INTEGER",
    "age" = "TEXT",
    "sex" = "TEXT",

    "isoyear" = "INTEGER",
    "isoweek" = "INTEGER",
    "isoyearweek" = "TEXT",
    "season" = "TEXT",
    "seasonweek" = "DOUBLE",

    "calyear" = "INTEGER",
    "calmonth" = "INTEGER",
    "calyearmonth" = "TEXT",

    "date" = "DATE",

    "tariffgroup_tag" = "TEXT",

    "consultations_icpc2group_n" = "INTEGER",
    "consultations_icpc2group_vs_all_pr100" = "DOUBLE",
    "consultations_icpc2group_vs_without_influenza_covid19_pr100" = "DOUBLE",
    "consultations_all_n" = "INTEGER",
    "consultations_without_influenza_covid19_n" = "INTEGER"
  ),
  keys = c(
    "granularity_time",
    "location_code",
    "date",
    "age",
    "sex",
    "tariffgroup_tag"
  ),
  indexes = list(
    "ind1" = c("isoyearweek"),
    "ind2" = c("granularity_geo", "age", "sex", "isoyear", "isoyearweek", "tariffgro
  ),
  validator_field_types = csdb::validator_field_types_csfmt_rts_data_v1,
  validator_field_contents = csdb::validator_field_contents_csfmt_rts_data_v1
)
```

Core Surveillance 9

Working with a database table

- `DBTable_v9$create_table()`
- `DBTable_v9$remove_table()`
- `DBTable_v9$insert_data()`
- `DBTable_v9$upsert_data()`
- `DBTable_v9$drop_all_rows()`
- `DBTable_v9$drop_rows_where()`
- `DBTable_v9$keep_rows_where()`
- `DBTable_v9$drop_all_rows_and_then_upsert_data()`
- `DBTable_v9$drop_all_rows_and_then_insert_data()`
- `DBTable_v9$tbl()`
- `DBTable_v9$print_dplyr_select()`
- `DBTable_v9$add_indexes()`
- `DBTable_v9$drop_indexes()`
- `DBTable_v9$confirm_indexes()`
- `DBTable_v9$nrow()`
- `DBTable_v9$info()`

Core Surveillance 9

Adding a new task

The only permitted loops are defined in the task definition, not in the action function (allows for very easy parallelization)

What will I do with my data?

How do I get my data?

What database tables can I access?

Run task in background as a RStudio job

Immediate parallelization over the plans

Partitioned tables avoid database lock

```
## + analyze_short_term_trends ----
# sc9::run_task_sequentially_as_rstudio_job_using_load_all("large_scale_surveillanc
global$ss$add_task(
  name_grouping = "large_scale_surveillance",
  name_action = "analyze",
  name_variant = "short_term_trends",
  cores = 20,
  plan_analysis_fn_name = NULL,
  for_each_plan = plnr::expand_list(
    icpc2group_tag = norsyss::icpc2$icpc2group_tag
  ),
  for_each_analysis = NULL,
  universal_argset = NULL,
  upsert_at_end_of_each_plan = FALSE,
  insert_at_end_of_each_plan = FALSE,
  action_fn_name = "norsyss.sc9::large_scale_surveillance_analyze_short_term_trends
  data_selector_fn_name = "norsyss.sc9::large_scale_surveillance_analyze_short_term
  tables = list(
    # input
    "anon_norsyss_data" = global$ss$partitionedtables$anon_norsyss_data,

    # output
    "anon_large_scale_surveillance_short_term_trends" = global$ss$partitionedtables
  )
)
```

Core Surveillance 9

Data selector function

Allows us to jump directly into the function, like it's a normal script

```
large_scale_surveillance_analyze_short_term_trends_data_selector = function(argset, tables){
  if(plnr::is_run_directly()){
    # global$ss$shortcut_get_plans_argsets_as_dt("large_scale_surveillance_analyze_short_term_trends")

    index_plan <- 1
    index_analysis <- 1

    argset <- global$ss$shortcut_get_argset("large_scale_surveillance_analyze_short_term_trends", inde
    tables <- global$ss$shortcut_get_tables("large_scale_surveillance_analyze_short_term_trends")
  }
```

Extracting the data that is required for the analysis

```
data <- tables$anon_norsyss_data$tables[[argset$icpc2group_tag]]$tbl() %>%
  sc9::mandatory_db_filter(
    granularity_time = "isoyearweek",
    granularity_time_not = NULL,
    granularity_geo = c("nation", "county"),
    granularity_geo_not = NULL,
    country_iso3 = NULL,
    location_code = NULL,
    age = c(
      "total",
      "000_004",
      "005_014",
      "015_019",
      "020_029",
      "030_064",
      "065_069",
      "070_079",
      "080p"
    ),
    age_not = NULL,
    sex = "total",
    sex_not = NULL
  ) %>%
```

Access table, name is from icpc2group_tag (comes from the plan!)

This is "the loop"!

Can also implement further filtering here, according to plan requirements

Action function

Can test entire task

Allows us to jump directly into the function, like it's a normal script

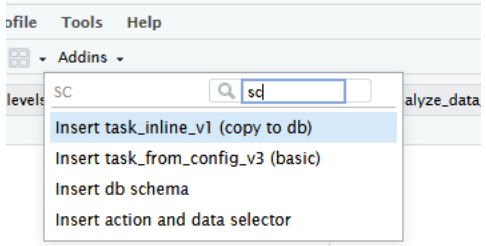
Code goes here. No loops!!

Insert to database. So easy!

```
large_scale_surveillance_analyze_short_term_trends_action <- function(data, argset, tables) {
  # sc9::run_task_sequentially_as_rstudio_job_using_load_all("large_scale_surveillance_analyze_short_term_trends")
  # To be run outside of rstudio: norsyss.sc9::global$ss$run_task("large_scale_surveillance_analyze_short_term_trends")

  if(plnr::is_run_directly()){
    # global$ss$shortcut_get_plans_argsets_as_dt("large_scale_surveillance_analyze_short_term_trends")

    index_plan <- 53
    index_analysis <- 1

    data <- global$ss$shortcut_get_data("large_scale_surveillance_analyze_short_term_trends", index_plan = index_plan)
    argset <- global$ss$shortcut_get_argset("large_scale_surveillance_analyze_short_term_trends", index_plan = index_pla
    tables <- global$ss$shortcut_get_tables("large_scale_surveillance_analyze_short_term_trends")

    # global$ss$partitionedtables$anon_norsyss_data$info()
    # global$ss$partitionedtables$anon_large_scale_surveillance_short_term_trends$info(collapse=T)
  }

  # code goes here
  # special case that runs before everything
  if(argset$first_analysis == TRUE){
    tables$anon_large_scale_surveillance_short_term_trends$drop_all_rows()
  }

  d <- copy(data$data)
  cstidy::set_csfmt_rts_data_v1(d)

  setnames(d, "consultations_icpc2group_n", "numerator_n")
  if(norsyss::icpc2[icpc2group_tag==argset$icpc2group_tag]$includes_influenza_covid19){
    setnames(d, "consultations_all_n", "denominator_n")
  } else {
    setnames(d, "consultations_without_influenza_covid19_n", "denominator_n")
  }

  x <- csalert::short_term_trend(
    d,
    numerator = "numerator_n",
    denominator = "denominator_n",
    prX = c(100),
    trend_isoyearweeks = 6,
    remove_last_isoyearweeks = 0,
    forecast_isoyearweeks = 2,
    numerator_naming_prefix = "generic",
    denominator_naming_prefix = "generic",
    statistics_naming_prefix = "universal",
    remove_training_data = TRUE
  )
  x <- x[numerator_forecasted_n_forecast  == FALSE]

  tables$anon_large_scale_surveillance_short_term_trends$insert_data(x)
```

Core Surveillance 9

How do I remember everything?




```r
# TASK_NAME ----
# tm_run_task("TASK_NAME")
sc::add_task(
  sc::task_from_config_v3(
    name_grouping = "TASK_GROUPING",
    name_action = "TASK_ACTION",
    name_variant = "TASK_VARIANT",
    cores = 1,
    plan_argset_fn_name = NULL, # "PACKAGE::TASK_NAME_plan_argset"
    for_each_plan = plnr::expand_list(
      x = 1
    ),
    for_each_argset = NULL,
    universal_argset = NULL,
    upsert_at_end_of_each_plan = FALSE,
    insert_at_end_of_each_plan = FALSE,
    action_fn_name = "PACKAGE::TASK_NAME_action",
    data_selector_fn_name = "PACKAGE::TASK_NAME_data_selector",
    schema = list(
      "SCHEMA_NAME" = sc::config$schemas$SCHEMA_NAME
    ),
    info = "This task does..."
  )
)
```

```r
# XGROUPX_XVARIANTX ----
sc::add_schema(
  schema = sc::Schema$new(
    db_table = "XGROUPX_XVARIANTX",
    db_config = sc::config$db_config,
    db_field_types =  c(
      "granularity_time" = "TEXT",
      "granularity_geo" = "TEXT",
      "location_code" = "TEXT",
      "border" = "INTEGER",
      "age" = "TEXT",
      "sex" = "TEXT",
      "isoyear" = "INTEGER",
      "isoweek" = "INTEGER",
      "isoyearweek" = "TEXT",
      "season" = "TEXT",
      "seasonweek" = "DOUBLE",
      "date" = "DATE",

      "XXXX" = "DOUBLE"
    ),
    db_load_folder = tempdir(),
    keys =  c(
      "granularity_time",
      "location_code",
      "date",
      "age",
      "sex"
    ),
    validator_field_types = sc::validator_
    validator_field_contents = sc::validat
    info = "This db table is used for..."
  )
)
```

Source on Save

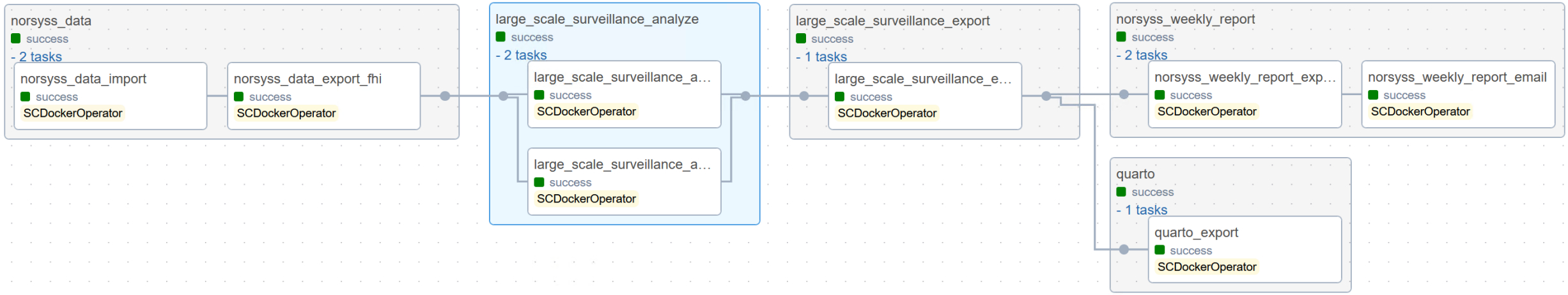
```r
# **** action **** ----
#' TASK_NAME (action)
#' @param data Data
#' @param argset Argset
#' @param schema DB Schema
#' @export
TASK_NAME_action <- function(data, argset, schema) {
  # tm_run_task("TASK_NAME")

  if(plnr::is_run_directly()){
    # sc::tm_get_plans_argsets_as_dt("TASK_NAME")

    index_plan <- 1
    index_argset <- 1

    data <- sc::tm_get_data("TASK_NAME", index_plan = index_plan)
    argset <- sc::tm_get_argset("TASK_NAME", index_plan = index_plan, index_argset = index_argset)
    schema <- sc::tm_get_schema("TASK_NAME")
  }

  # code goes here
}

# **** data_selector **** ----
#' TASK_NAME (data selector)
#' @param argset Argset
#' @param schema DB Schema
#' @export
TASK_NAME_data_selector = function(argset, schema){
  if(plnr::is_run_directly()){
    # sc::tm_get_plans_argsets_as_dt("TASK_NAME")

    index_plan <- 1

    argset <- sc::tm_get_argset("TASK_NAME", index_plan = index_plan)
    schema <- sc::tm_get_schema("TASK_NAME")
  }

  # The database schemas can be accessed here
  d <- schema$SCHEMA_NAME$dplyr_tbl() %>%
    dplyr::collect() %>%
    as.data.table()

  # The variable returned must be a named list
  retval <- list(
    "NAME" = d
```

77:1 **** functions ****

Environment History Connections Build Git Tutorial

Orchestration and tasks

Airflow + Core Surveillance 9



Which plan/analysis?

Core Surveillance 9

```
> global$ss$shortcut_get_plans_argsets_as_dt("large_scale_surveillance_analyze_short_term_trends")
    index_plan index_analysis **universal** **plan**         icpc2group_tag **analysis**
 1:          1              1             *        *                    a02            *
 2:          2              1             *        *                    a03            *
 3:          3              1             *        *                    a04            *
 4:          4              1             *        *                    a05            *
 5:          5              1             *        *                    a71            *
 6:          6              1             *        *                    a72            *
 7:          7              1             *        *                    a73            *
 8:          8              1             *        *                    a74            *
 9:          9              1             *        *                    a75            *
10:         10              1             *        *                    a76            *
11:         11              1             *        *                    a77            *
12:         12              1             *        *                    a78            *
13:         13              1             *        *                    b02            *
14:         14              1             *        *                    b70            *
15:         15              1             *        *                    b71            *
16:         16              1             *        *        gastroenteritis            *
17:         17              1             *        *                    d01            *
18:         18              1             *        *                    d02            *
19:         19              1             *        *                    d06            *
20:         20              1             *        *                    d08            *
21:         21              1             *        *                    d09            *
22:         22              1             *        *                    d10            *
23:         23              1             *        *                    d11            *
24:         24              1             *        *                    d14            *
25:         25              1             *        *                    d18            *
26:         26              1             *        *                    d25            *
```

Run as Rstudio job

```
#' @export
large_scale_surveillance_export_figures_and_tables_action <- function(data, argset, tables) {
  # sc9::run_task_sequentially_as_rstudio_job_using_load_all("large_scale_surveillance_export_fi
  # To be run outside of rstudio: norsyss.sc9::global$ss$run_task("large_scale_surveillance_expo

  if(plnr::is_run_directly()){
    # global$ss$shortcut_get_plans_argsets_as_dt("large_scale_surveillance_export_figures_and_ta

    index_plan <- 41
    index_analysis <- 1

    data <- global$ss$shortcut_get_data("large_scale_surveillance_export_figures_and_tables", in
    argset <- global$ss$shortcut_get_argset("large_scale_surveillance_export_figures_and_tables"
```

norsyss.sc9 | R 4.3.0

Console | Terminal | Background Jobs

Start Background Job

large_scale_surveillance_export_figures_... 0:34

norsyss.sc9 | R 4.3.0

Console | Terminal | Background Jobs

large_scale_surveillance_export_figures_and_tables 0:50

```
**devtools::load_all**
ℹ Loading norsyss.sc9
**run_task**
norsyss.sc9 2023.6.28
sc9 2023.8.1
task: large_scale_surveillance_export_figures_and_tables
Running task=large_scale_surveillance_export_figures_and_tables with plans=1424 and analyses=1424
plans=sequential, argset=sequential with cores=1
   2 / 1424 =   0%     427m ->   1m = 427.2m
```

Daily data export

Data selector function

```yaml
---
- folder: sc9::path("fra_norsyss_til_luftveisovervaking", "ukesrapport")
  filename: "ukesrapport_v1_{lubridate::today()}.RDS"
  data:
    - name: "bottom_of_disease_pyramid"
      granularity_geo:
        - "nation"
      granularity_time:
        - "isoyearweek"
      age:
        - "total"
      sex:
        - "total"
      tariffgroup_tag:
        - "fe"
      icpc2group_tag:
        - "r992"
        - "r991"
        - "covid19"
        - "respiratory_infections"
    - name: "respiratory_comparison"
      granularity_geo:
        - "nation"
      granularity_time:
        - "isoyearweek"
      age:
        - "total"
      sex:
        - "total"
      tariffgroup_tag:
        - "fe"
        - "fes"
      icpc2group_tag:
        - "covid19"
        - "r27"
        - "r80"
        - "r74"
        - "respiratory_infections"
```

```r
retval <- list()
for(i in seq_along(argset$data)){
  di <- argset$data[[i]]
  d_partition <- vector("list", length = length(di$icpc2group_tag))
  for(j in seq_along(di$icpc2group_tag)){
    d_partition[[j]] <- tables$anon_norsyss_data$tables[[di$icpc2group_tag[j]]]$tbl() %>%
      sc9::mandatory_db_filter(
        granularity_time = di$granularity_time,
        granularity_time_not = NULL,
        granularity_geo = di$granularity_geo,
        granularity_geo_not = NULL,
        country_iso3 = NULL,
        location_code = NULL,
        age = di$age,
        age_not = NULL,
        sex = di$sex,
        sex_not = NULL
      ) %>%
      dplyr::filter(tariffgroup_tag %in% !!di$tariffgroup_tag) %>%
```

Action function

```r

tryCatch({
  folder <- eval(parse(text = argset$folder))
  file <- eval(parse(text = paste0("glue::glue(\"",argset$file,"\")")))

  print(folder)
  print(file)

  dir.create(folder, showWarnings = FALSE, recursive = TRUE)

  if(stringr::str_detect(argset$filename, "RDS$")){
    saveRDS(data, fs::path(folder, file))
  } else if(stringr::str_detect(argset$filename, "csv$")){
    hash_index <- which(names(data)=="hash")
    data <- rbindlist(data[-hash_index])
    fwrite(data, fs::path(folder, file))
  }
}, warning = function(w){
  print("FAILED!")
}, error = function(e){
  print("FAILED!")
})

```

Figures

ntitled1 | 04_tasks.R | figures_and_tables.yaml | 03_db_tables.R | epistatusmøte.qmd

```yaml
- hash_yaml: f82f884f277c90c67e578cc25b1f93c9
  hash_tasks: c3ac52a61294592834f59a1b4ac8021a
- task_name: large_scale_surveillance_export_figures_and_tables
  figures:
  - name_1: comparisons
    name_2: seasons
    name_3: argset$location_code
    name_4: argset$icpc2group_tag
    filename: '{argset$location_code}_{argset$icpc2group_tag}_comparisons_seasons.png'
    directorypath_today: sc9::path("norsyss_internal_output", "large_scale_surveillance",
      "comparisons", lubridate::today())
    directorypath_latest: sc9::path("norsyss_internal_output", "large_scale_surveillance",
      "comparisons", "_latest")
  - name_1: short_term_trends
    name_2: epicurve
    name_3: argset$location_code
    name_4: argset$icpc2group_tag
    filename: '{argset$location_code}_{argset$icpc2group_tag}_short_term_trends_epicurve.png'
    directorypath_today: sc9::path("norsyss_internal_output", "large_scale_surveillance",
      "short_term_trends", lubridate::today())
    directorypath_latest: sc9::path("norsyss_internal_output", "large_scale_surveillance",
      "short_term_trends", "_latest")
```

```
1420:                    TRUE
1421:                    TRUE
1422:
1423:
1424:
>   # code goes here
>   figures <- global$figur
> figures$short_term_trends$
```

epicurve
epicurve_by_age
map
tiles_by_locationumbrella_age

```
 3:                      TRUE
 4:                      TRUE
 5:                      TRUE
---
420:                     TRUE
421:                     TRUE
422:                     TRUE
423:                     TRUE
424:                     TRUE
   # code goes here
   figures <- global$figures[["larg
 figures$short_term_trends$epicurve$
```

nation_nor
county_nor42
county_nor32
county_nor33
county_nor56
county_nor34
county_nor15
county_nor18
county_nor03

```
 3:                      TRUE
 4:                      TRUE
 5:                      TRUE
---
420:                     TRUE
421:                     TRUE
422:                     TRUE
423:                     TRUE
424:                     TRUE
   # code goes here
   figures <- global$figures[["large_scale_su
 figures$short_term_trends$epicurve$nation_nor$
> figures$short_term_trends$epicurve$nation_nor$a02$today
[1] "~/_mnt/n/norsyss_internal_output/_interactive/large_scale_surveillance/shor
m_trends_epicurve.png"
>
```

a02
a03
a04
a05
a71
a72
a73
a74
a75

Figures in quarto

````
```{r}
#| echo: false
if(params$load_all == "true" | interactive()) devtools::load_all("~/norsyss.sc9")

figures <- norsyss.sc9::global$figures$large_scale_surveillance_export_figures_and_tables
```

## Covid-19-konsultasjonar: Trendar

:::: {.columns}

::: {.column width="50%"}

### Alle aldrar

```{r}
knitr::include_graphics(norsyss.sc9::global$figures$large_scale_surveillance_export_figures_and_tables$short_term_trends$epicurve$nation_nor$covid19$latest)
```

:::

::: {.column width="50%"}

### Samanlikn sesongar

```{r}
knitr::include_graphics(norsyss.sc9::global$figures$large_scale_surveillance_export_figures_and_tables$comparisons$seasons$nation_nor$covid19$latest)
```
````

Statistics on database tables

Core Surveillance 9

```
> tables$anon_norsyss_data$info()
                                      table_name     nrow size_total_gb size_data_gb size_index_gb
 1:                anon_norsyss_data_PARTITION_a02 15693975         11.92         5.84          6.06
 2:                anon_norsyss_data_PARTITION_a03 15693975         11.92         5.85          6.07
 3:                anon_norsyss_data_PARTITION_a04 15693975         10.17         5.86          4.29
 4:                anon_norsyss_data_PARTITION_a05 15693975         11.89         5.83          6.05
 5:                anon_norsyss_data_PARTITION_a71 15693975         11.87         5.81          6.06
 6:                anon_norsyss_data_PARTITION_a72 15693975         11.92         5.85          6.06
 7:                anon_norsyss_data_PARTITION_a73 15693975         11.96         5.88          6.07
 8:                anon_norsyss_data_PARTITION_a74 15693975         11.93         5.85          6.07
 9:                anon_norsyss_data_PARTITION_a75 15693975         11.88         5.82          6.05
10:                anon_norsyss_data_PARTITION_a76 15693975         11.89         5.83          6.06
11:                anon_norsyss_data_PARTITION_a77 15693975         11.94         5.86          6.06
12:                anon_norsyss_data_PARTITION_a78 15693975         11.85         5.79          6.05
13:                anon_norsyss_data_PARTITION_b02 15693975         11.91         5.84          6.06
14:                anon_norsyss_data_PARTITION_b70 15693975         11.87         5.81          6.06
15:                anon_norsyss_data_PARTITION_b71 15693975         11.90         5.83          6.07
16:              anon_norsyss_data_PARTITION_covid19 15693975         12.12         6.04          6.07
                   anon_norsyss_data_PARTITION_d01 15693975         11.81         5.75          6.05
                   anon_norsyss_data_PARTITION_d02 15693975         11.87         5.80          6.06
```

```
> sc9::get_config_tables_last_updated()
                         table_name       date            datetime
  1:           anon_euromomo_results 2023-12-05 2023-12-05 10:52:16
  2:           anon_example_income_a 2023-03-29 2023-03-29 14:22:37
  3:           anon_example_income_b 2023-03-29 2023-03-29 14:22:38
  4:       anon_example_weather_data 2023-05-16 2023-05-16 01:16:54
  5:    anon_example_weather_rawdata 2023-05-16 2023-05-16 01:14:24
 ---
269: anon_norsyss_data_PARTITION_s75 2023-12-05 2023-12-05 01:30:15
270: anon_norsyss_data_PARTITION_s76 2023-12-05 2023-12-05 01:29:07
271: anon_norsyss_data_PARTITION_s84 2023-12-05 2023-12-05 01:29:08
272: anon_norsyss_data_PARTITION_s95 2023-12-05 2023-12-05 01:27:45
273: anon_norsyss_data_PARTITION_s99 2023-12-05 2023-12-05 01:28:09
>
```

Statistics on tasks

Core Surveillance 9

```
> sc9::get_config_tasks_stats()
                                task sc_version implementation_version cores_n plans_n analyses_n start_date      start_datetime  stop_date
   1:  euromomo_export_figures_and_tables   2023.8.1             2023.11.17       1       1          1 2023-11-29 2023-11-29 23:02:06 2023-11-29
   2:  euromomo_export_figures_and_tables   2023.8.1             2023.11.17       1       1          1 2023-11-29 2023-11-29 23:02:34 2023-11-29
   3:  euromomo_export_figures_and_tables   2023.8.1             2023.11.17       1       1          1 2023-11-29 2023-11-29 23:13:05 2023-11-29
   4:  euromomo_export_figures_and_tables   2023.8.1             2023.11.17       1       1          1 2023-11-30 2023-11-30 01:07:45 2023-11-30
   5:  euromomo_export_figures_and_tables   2023.8.1             2023.11.17       1       1          1 2023-12-01 2023-12-01 01:07:20 2023-12-01
  ---
1839: weather_download_and_import_rawdata   2023.5.3               2023.5.4       1     356        356 2023-05-12 2023-05-12 01:05:10 2023-05-12
1840: weather_download_and_import_rawdata   2023.5.3               2023.5.4       1     356        356 2023-05-13 2023-05-13 01:05:09 2023-05-13
1841: weather_download_and_import_rawdata   2023.5.3               2023.5.4       1     356        356 2023-05-14 2023-05-14 01:05:09 2023-05-14
1842: weather_download_and_import_rawdata   2023.5.3               2023.5.4       1     356        356 2023-05-15 2023-05-15 01:05:10 2023-05-15
1843: weather_download_and_import_rawdata   2023.5.3               2023.5.4       1     356        356 2023-05-16 2023-05-16 01:05:09 2023-05-16
           stop_datetime runtime_minutes ram_all_cores_mb ram_per_core_mb    status
   1: 2023-11-29 23:02:11            0.07            319.1           319.1 succeeded
   2: 2023-11-29 23:02:38            0.07            319.1           319.1 succeeded
   3: 2023-11-29 23:13:10            0.08            311.2           311.2 succeeded
   4: 2023-11-30 01:07:49            0.08            311.2           311.2 succeeded
   5: 2023-12-01 01:07:26            0.09            311.2           311.2 succeeded
  ---
1839: 2023-05-12 01:13:56            8.77            173.8           173.8 succeeded
1840: 2023-05-13 01:13:45            8.59            173.9           173.9 succeeded
1841: 2023-05-14 01:13:57            8.79            174.0           174.0 succeeded
1842: 2023-05-15 01:12:03            6.88            174.0           174.0 succeeded
1843: 2023-05-16 01:14:26            9.27            174.1           174.1 succeeded
>
```

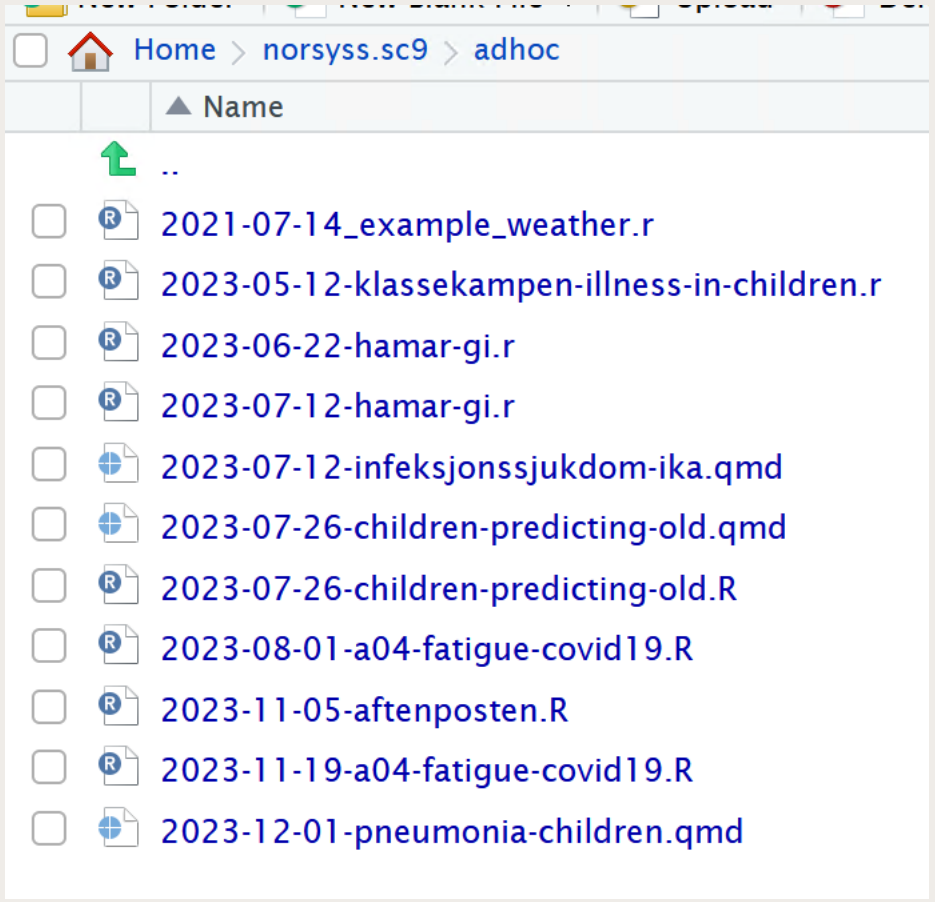
Hashing data

```
$hash
$hash$current
[1] "9673ecd375f9f6b75d83099d1ad1af2d"

$hash$current_elements
$hash$current_elements$data
[1] "78e73cf8cbf4a70462058cbb104b5225"

$hash$current_elements$hdata
[1] "6daa507563518535df6a942e95368039"

$hash$current_elements$extraction_date
[1] "c4682d0a30ce9f07fba1108e67fdde92"
```

```
> data
$data
        [redacted]  age
     1:             67
     2:             87
     3:             80
     4:             86
     5:             88
    ---
679529:             67
679530:             73
679531:             84
679532:             74
679533:             74

$hdata
           date closed
  1: 2000-04-16      1
  2: 2000-04-20      1
  3: 2000-04-21      1
  4: 2000-04-23      1
  5: 2000-04-24      1
 ---
304: 2023-05-18      1
305: 2023-05-28      1
306: 2023-05-29      1
307: 2023-12-25      1
308: 2023-12-26      1

$extraction_date
[1] "2023-12-05"
```

```
> sc9::get_config_data_hash_for_each_plan(task="euromomo_run_analysis")
                   task index_plan     element_tag       date            datetime                    element_hash                     all_hash
 1: euromomo_run_analysis          1            data 2023-11-29 2023-11-29 23:15:54 8bd57ea1c55a53b9065203f60631ddd0 678f53efd8f7f684581dcd52179454b4
 2: euromomo_run_analysis          1            data 2023-11-30 2023-11-30 01:07:33 8bd57ea1c55a53b9065203f60631ddd0 a638c3b74950781e6b542ac15efea211
 3: euromomo_run_analysis          1            data 2023-12-01 2023-12-01 01:07:10 8bd57ea1c55a53b9065203f60631ddd0 69a83f0f200f2478369a96051403f104
 4: euromomo_run_analysis          1            data 2023-12-02 2023-12-02 01:07:14 8bd57ea1c55a53b9065203f60631ddd0 c2b0be9cbadc8710faf1f5b9241e85c7
 5: euromomo_run_analysis          1            data 2023-12-03 2023-12-03 01:07:09 8bd57ea1c55a53b9065203f60631ddd0 4051df7e200bf53a035f16be8b8ba245
 6: euromomo_run_analysis          1            data 2023-12-04 2023-12-04 01:07:11 8bd57ea1c55a53b9065203f60631ddd0 02d9310e12a9bbeedf8eead5685bf889
 7: euromomo_run_analysis          1            data 2023-12-05 2023-12-05 01:07:08 8bd57ea1c55a53b9065203f60631ddd0 73fda6cd3850ee68cebd5f0d9ac2b079
 8: euromomo_run_analysis          1            data 2023-12-05 2023-12-05 10:52:16 78e73cf8cbf4a70462058cbb104b5225 9673ecd375f9f6b75d83099d1ad1af2d
 9: euromomo_run_analysis          1 extraction_date 2023-11-29 2023-11-29 23:15:54 ce5b6ea933af93d0d416d42c1722ee5a 678f53efd8f7f684581dcd52179454b4
10: euromomo_run_analysis          1 extraction_date 2023-11-30 2023-11-30 01:07:33 390a6d02282ebdfee6bd4bf6525fc9ea a638c3b74950781e6b542ac15efea211
11: euromomo_run_analysis          1 extraction_date 2023-12-01 2023-12-01 01:07:10 6af0581db4297831cce37bd8362ac0a4 69a83f0f200f2478369a96051403f104
12: euromomo_run_analysis          1 extraction_date 2023-12-02 2023-12-02 01:07:14 c85e077afcf865c00ae9a07ed54f8e4f c2b0be9cbadc8710faf1f5b9241e85c7
13: euromomo_run_analysis          1 extraction_date 2023-12-03 2023-12-03 01:07:09 7cd6380e7b74851330a7f2acf3595e57 4051df7e200bf53a035f16be8b8ba245
14: euromomo_run_analysis          1 extraction_date 2023-12-04 2023-12-04 01:07:11 2c429d0deb521a543b4b6f316377d64c 02d9310e12a9bbeedf8eead5685bf889
15: euromomo_run_analysis          1 extraction_date 2023-12-05 2023-12-05 01:07:08 c4682d0a30ce9f07fba1108e67fdde92 73fda6cd3850ee68cebd5f0d9ac2b079
16: euromomo_run_analysis          1 extraction_date 2023-12-05 2023-12-05 10:52:16 c4682d0a30ce9f07fba1108e67fdde92 9673ecd375f9f6b75d83099d1ad1af2d
17: euromomo_run_analysis          1           hdata 2023-11-29 2023-11-29 23:15:54 6daa507563518535df6a942e95368039 678f53efd8f7f684581dcd52179454b4
18: euromomo_run_analysis          1           hdata 2023-11-30 2023-11-30 01:07:33 6daa507563518535df6a942e95368039 a638c3b74950781e6b542ac15efea211
19: euromomo_run_analysis          1           hdata 2023-12-01 2023-12-01 01:07:10 6daa507563518535df6a942e95368039 69a83f0f200f2478369a96051403f104
20: euromomo_run_analysis          1           hdata 2023-12-02 2023-12-02 01:07:14 6daa507563518535df6a942e95368039 c2b0be9cbadc8710faf1f5b9241e85c7
21: euromomo_run_analysis          1           hdata 2023-12-03 2023-12-03 01:07:09 6daa507563518535df6a942e95368039 4051df7e200bf53a035f16be8b8ba245
22: euromomo_run_analysis          1           hdata 2023-12-04 2023-12-04 01:07:11 6daa507563518535df6a942e95368039 02d9310e12a9bbeedf8eead5685bf889
23: euromomo_run_analysis          1           hdata 2023-12-05 2023-12-05 01:07:08 6daa507563518535df6a942e95368039 73fda6cd3850ee68cebd5f0d9ac2b079
24: euromomo_run_analysis          1           hdata 2023-12-05 2023-12-05 10:52:16 6daa507563518535df6a942e95368039 9673ecd375f9f6b75d83099d1ad1af2d
                   task index_plan     element_tag       date            datetime                    element_hash                     all_hash
>
```

Ad-hoc analyses



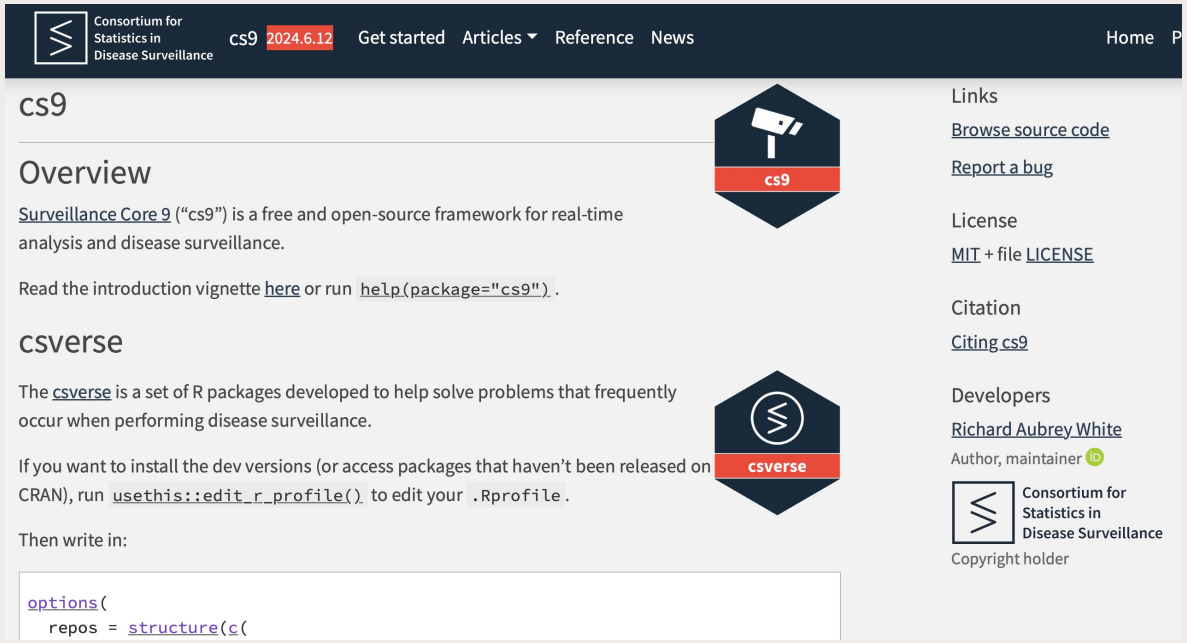

```r
devtools::load_all()

retval <- global$ss$partitionedtables$anon_norsyss_data$tables[["a04"]]$tbl() %>%
  sc9::mandatory_db_filter(
    granularity_time = "isoyearweek",
    granularity_time_not = NULL,
    granularity_geo = c("nation"),
    granularity_geo_not = NULL,
    country_iso3 = NULL,
    location_code = NULL,
    age = c(
      "total",
      "000_004",
      "005_014",
      "015_019",
      "020_029",
      "030_064",
      "065_069",
      "070_079",
      "080p"
    ),
    age_not = NULL,
    sex = "total",
    sex_not = NULL
  ) %>%
  dplyr::filter(tariffgroup_tag == "fe") %>%
```

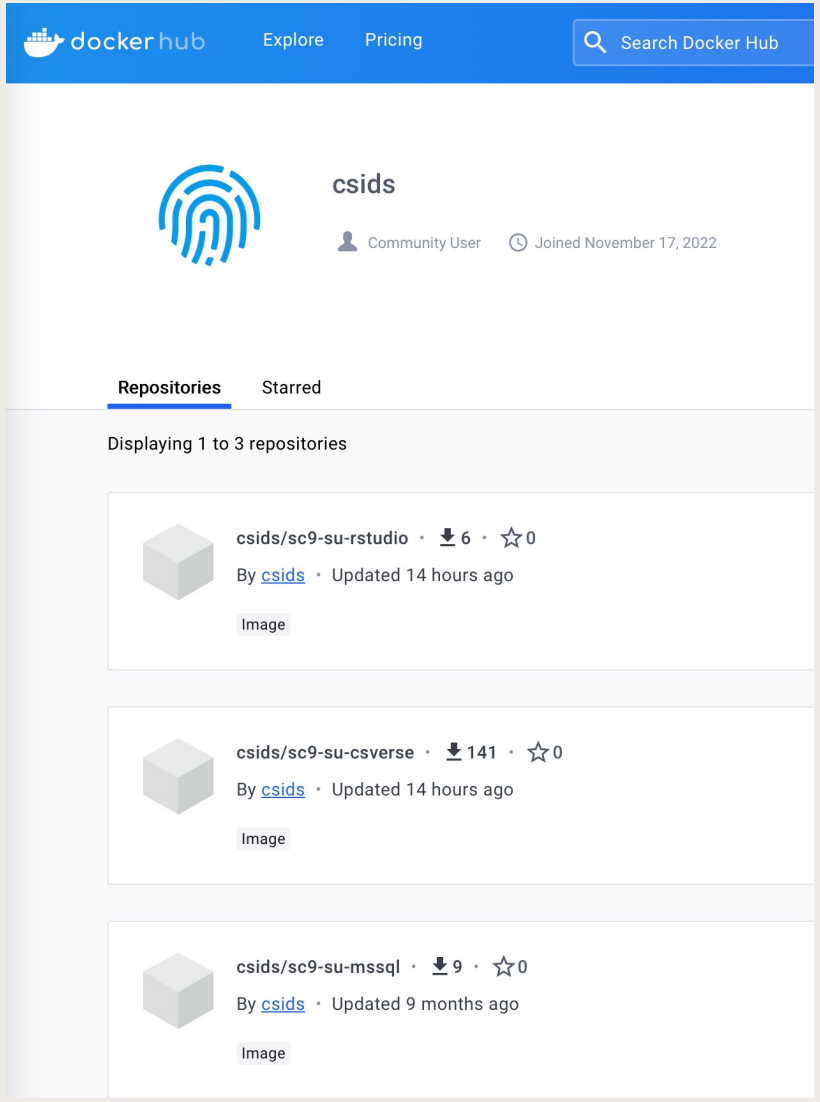
```r
retval[, type := "Weeks 1-39"]
retval[isoweek >= 40, type := "Weeks 40-52"]
pd <- retval[, .(
  consultations_icpc2group_n = sum(consultations_icpc2group_n)
), keyby = .(
  age,
  isoyear,
  type
)]
```

Use it yourself?

www.csids.no/cs9

hub.docker.com/u/csids





Consortium for
Statistics in
Disease Surveillance

NIPH
Norwegian Institute of Public Health

Questions?