

# Package: naomi (via r-universe)

December 5, 2024

**Title** Naomi Model for Subnational HIV Estimates

**Version** 2.10.4

**Description** This package implements the Naomi model for subnational HIV estimates.

**Depends** R (>= 3.5.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Additional\_repositories** <https://mrc-ide.r-universe.dev>,  
<https://duckdb.r-universe.dev>

**Imports** DBI, Matrix (>= 1.6.5), R6, TMB, assertthat, brio, data.tree, dplyr (>= 1.1.0), eppasm (>= 0.7.1), gt, first90 (>= 1.7.1), forcats, fs, ggplot2, magrittr, methods, mvtnorm, naomi.options (>= 1.3.0), openxlsx, plotly, prettyunits, qs, readr (>= 2.0.1), rlang, rmarkdown, sf, spdep (>= 1.1), stringr, tidyr (>= 1.3.0), tidyselect (>= 1.0.0), traduire, utils, withr, writexl, yaml, zip, zoo

**Suggests** DiagrammeR, covr, datamodelr, duckdb (>= 0.10.0), here, knitr, lubridate, mockery, mockr, readxl, rvest, scales, testthat (>= 2.1.0), tibble, tidyverse, viridis

**VignetteBuilder** knitr

**LinkingTo** RcppEigen, TMB

**Remotes** bergant/datamodelr, mrc-ide/naomi.options,  
first90=github::mrc-ide/first90release

**Config/testthat/edition** 3

**Config/testthat/parallel** true

**Config/pak/sysreqs** libgdal-dev gdal-bin libgeos-dev make libicu-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev libnode-dev libx11-dev zlib1g-dev

**Repository** <https://mrc-ide.r-universe.dev>

**RemoteUrl** <https://github.com/mrc-ide/naomi>

**RemoteRef** master

**RemoteSha** c71c907374ea7110f971904f98526ef33b6a724a8

## Contents

add_output_labels . . . . .	4
age_bar_plotly . . . . .	4
aggregate_anc . . . . .	5
aggregate_art . . . . .	5
align_inputs_outputs . . . . .	6
anc_testing_prev_mf . . . . .	6
artnum_mf . . . . .	7
assert_pjnz_shiny90 . . . . .	7
bar_plotly . . . . .	8
build_hierarchy_label . . . . .	8
calculate_prevalence_art_coverage . . . . .	9
calibrate_outputs . . . . .	9
cmc_date . . . . .	11
create_adj_matrix . . . . .	12
create_areas . . . . .	13
create_area_aggregation . . . . .	13
create_edge_list . . . . .	14
cut_naomi_age_group . . . . .	14
data_type_labels . . . . .	15
demo_anc_testing . . . . .	15
demo_area_boundaries . . . . .	17
demo_area_hierarchy . . . . .	18
demo_area_levels . . . . .	18
demo_population_agesex . . . . .	19
demo_survey_biomarker . . . . .	19
demo_survey_clusters . . . . .	20
demo_survey_hiv_indicators . . . . .	20
demo_survey_individuals . . . . .	21
demo_survey_meta . . . . .	21
demo_survey_regions . . . . .	22
disaggregate_0to4_outputs . . . . .	22
district_barplot . . . . .	23
dropdown_buttons . . . . .	24
extract_pjnz_naomi . . . . .	24
extract_pjnz_program_data . . . . .	25
extract_shiny90_age_sex . . . . .	25
fit_tmb . . . . .	26
get_age_groups . . . . .	27
get_anc_metadata . . . . .	27
get_area_collection . . . . .	28

get_art_metadata . . . . .	29
get_five_year_age_groups . . . . .	29
get_metadata . . . . .	30
get_meta_indicator . . . . .	30
get_plotting_metadata . . . . .	31
get_plot_type_column_metadata . . . . .	31
get_spec_aggr_interpolation . . . . .	32
hintr_calibrate . . . . .	32
hintr_calibrate_plot . . . . .	33
hintr_comparison_plot . . . . .	33
hintr_prepare_agyw_download . . . . .	34
hintr_prepare_coarse_age_group_download . . . . .	34
hintr_prepare_comparison_report_download . . . . .	35
hintr_prepare_spectrum_download . . . . .	35
hintr_prepare_summary_report_download . . . . .	36
hintr_run_model . . . . .	36
hintr_validate_programme_data . . . . .	38
Imports . . . . .	38
interpolate_population_agesex . . . . .	38
log_linear_interp . . . . .	39
map_outputs . . . . .	40
naomi_model_frame . . . . .	41
naomi_objective_function_r . . . . .	43
naomi_output_frame . . . . .	44
output_naomi_warning . . . . .	44
output_package . . . . .	45
pop_pyramid_outputs . . . . .	45
prepare_input_time_series_anc . . . . .	46
prepare_input_time_series_art . . . . .	47
prepare_spectrum_naomi_comparison . . . . .	47
prepare_tmb_inputs . . . . .	48
quarter_year_labels . . . . .	48
read_hintr_output . . . . .	50
read_population . . . . .	50
read_spectrum_projection_name . . . . .	51
read_spectrum_region_code . . . . .	51
read_spectrum_region_name . . . . .	52
report_tmb . . . . .	53
sample_tmb . . . . .	53
save_output . . . . .	54
save_output_package . . . . .	55
scale_gmrf_precision . . . . .	56
scatter_plotly . . . . .	56
select_naomi_data . . . . .	57
spread_areas . . . . .	59
subset_naomi_output . . . . .	60
subset_output_package . . . . .	61
survey_mf . . . . .	62

th_map . . . . .	62
validate_model_options . . . . .	63
write_datapack_csv . . . . .	63
write_navigator_checklist . . . . .	64

<b>Index</b>	<b>65</b>
--------------	-----------

---

add_output_labels	<i>Add labels to output indicators</i>
-------------------	--

---

### Description

Add labels to output indicators

### Usage

```
add_output_labels(naomi_output, geometry = FALSE)
```

### Arguments

naomi_output	Naomi output object.
geometry	logical whether to include geometry from meta_area.

### Value

Labelled output indicators

---

age_bar_plotly	<i>Plotly barplot comparing age distribution of survey inputs and naomi outputs</i>
----------------	---

---

### Description

Plotly barplot comparing age distribution of survey inputs and naomi outputs

### Usage

```
age_bar_plotly(df, ind, quarter)
```

### Arguments

df	Inputs_outputs dataframe containing matched model estimates and data inputs.
ind	Indicator filter.
quarter	Calendar quarter filter.

---

aggregate_anc	<i>Aggregate ANC data according to area hierarchy</i>
---------------	---

---

**Description**

Take ANC and shape file paths or data files and aggregate art\_current according to area hierarchy provided

**Usage**

```
aggregate_anc(anc, shape)
```

**Arguments**

anc	Path to file containing ANC data or ANC data object
shape	Path to file containing geojson areas data or areas data object

**Value**

Aggregated ANC data containing columns area\_id, area\_name, area\_level, area\_level\_label, sex, age\_group, time\_period, year, quarter, calendar\_quarter, anc\_clients, anc\_known\_neg, anc\_already\_art, anc\_tested and anc\_tested\_pos, births\_clients\_ratio

---

aggregate_art	<i>Aggregate ART data according to area hierarchy</i>
---------------	---

---

**Description**

Take ART and shape file paths or files and aggregate art\_current according to area hierarchy provided

**Usage**

```
aggregate_art(art, shape)
```

**Arguments**

art	Path to file containing ART data or ART data object
shape	Path to file containing geojson areas data or areas data object

**Value**

Aggregated ART data containing columns area\_id, area\_name, area\_level, area\_level\_label, parent\_area\_id, sex, age\_group, time\_period, year, quarter, calendar\_quarter and art\_current

align\_inputs\_outputs *Align model data inputs and model estimates*

---

**Description**

Align model data inputs and model estimates

**Usage**

```
align_inputs_outputs(naomi_data, indicators, meta_area)
```

**Arguments**

naomi\_data      Naomi object of class "naomi\_data" and "naomi\_mf"  
indicators      Naomi indicators created by extract\_indicators()  
meta\_areas

---

anc\_testing\_prev\_mf *Prepare Model Frames for Programme Datasets*

---

**Description**

Prepare Model Frames for Programme Datasets

**Usage**

```
anc_testing_prev_mf(year, anc_model_mf)  
anc_testing_artcov_mf(year, anc_model_mf)  
anc_testing_clients_mf(year, anc_model_mf)
```

**Arguments**

year              Calendar year  
anc\_testing      ART data frame  
naomi\_mf          Naomi model frame

**Value**

Calculated prevalence

---

artnum_mf	<i>Build artnum model frame</i>
-----------	---------------------------------

---

**Description**

Build artnum model frame

**Usage**

```
artnum_mf(calendar_quarter, art_number, naomi_mf)
```

**Arguments**

calendar\_quarter  
Calendar quarter

**Details**

Number on ART at desired quarter are linearly interpolated within the dataset. If the desired quarter is before the earliest data, the first value may be carried back by up to one year (four quarters). Data are never carried forward

---

assert_pjnz_shiny90	<i>Check whether PJNZ contains .shiny90 file</i>
---------------------	--

---

**Description**

Check whether PJNZ contains .shiny90 file

**Usage**

```
assert_pjnz_shiny90(pjnz)
```

**Arguments**

pjnz           file path to PJNZ

**Value**

Logical whether PJNZ file contains a .shiny90 file

---

bar_plotly	<i>Plotly barplot comparing geographical distribution of data inputs and naomi outputs</i>
------------	--

---

**Description**

Plotly barplot comparing geographical distribution of data inputs and naomi outputs

**Usage**

```
bar_plotly(df, ind, quarter, age_disag = "Y015_049", sex_disag = "both")
```

**Arguments**

df	Inputs_outputs dataframe containing matched model estimates and data inputs.
ind	Indicator filter.
quarter	Calendar quarter filter.
age_disag	Age group filter.
sex_disag	Sex filter.

---

build_hierarchy_label	<i>Return the translated label &amp; description for a set of plot types</i>
-----------------------	--

---

**Description**

Return the translated label & description for a set of plot types

**Usage**

```
build_hierarchy_label(meta_areas)
```

**Arguments**

meta_areas	dataframe containing
------------	----------------------

**Value**

For each plot type the label and description as a list of lists containing id, label and description



---

`calculate_prevalence_art_coverage`*Calculate prevalence and ART coverage from ANC input data*

---

**Description**

Calculate prevalence and ART coverage from ANC input data

**Usage**

```
calculate_prevalence_art_coverage(data)
```

**Arguments**

`data`                    Data frame of input ANC data

**Value**

Data frame with prevalence and ART coverage

---

`calibrate_outputs`        *Calibrate naomi model outputs*

---

**Description**

Calibrate naomi model outputs

**Usage**

```
calibrate_outputs(  
  output,  
  naomi_mf,  
  spectrum_plhiv_calibration_level,  
  spectrum_plhiv_calibration_strat,  
  spectrum_artnum_calibration_level,  
  spectrum_artnum_calibration_strat,  
  spectrum_aware_calibration_level,  
  spectrum_aware_calibration_strat,  
  spectrum_infections_calibration_level,  
  spectrum_infections_calibration_strat,  
  calibrate_method = "logistic"  
)
```

**Arguments**

output	Naomi model output package produced by <code>output_package()</code> .
naomi_mf	Naomi model frame, objective of class <code>naomi_mf</code> .
spectrum_plhiv_calibration_level	Level to calibrate PLHIV, see details.
spectrum_plhiv_calibration_strat	Age/sex stratification to calibrate PLHIV, see details.
spectrum_artnum_calibration_level	Level to calibrate number on ART, see details.
spectrum_artnum_calibration_strat	Age/sex stratification to calibrate number on ART, see details.
spectrum_aware_calibration_level	Level to calibrate number aware of HIV positive status, see details.
spectrum_aware_calibration_strat	Age/sex stratification to calibrate number aware of HIV positive status, see details.
spectrum_infections_calibration_level	Level to calibrate number infections of HIV positive status, see details.
spectrum_infections_calibration_strat	Age/sex stratification to calibrate number infections of HIV positive status, see details.
calibrate_method	Calibration method, either "logistic" (default) or "proportional".

**Details**

The following indicators are calibrated:

- plhiv
- art\_current\_residents
- unaware\_plhiv\_num
- infections
- art\_current (attending)
- aware\_plhiv\_num
- untreated\_plhiv\_num
- prevalence
- art\_coverage
- aware\_plhiv\_prop
- incidence
- plhiv\_attend
- aware\_plhiv\_attend

Steps in the calibration:

1. Aggregate Spectrum totals to specified stratification by level/sex/age to calculate the target totals within each stratification.
2. Adjust fine area/sex/age-group mean values to match targeted total using either "logistic" or "proportional" scaling.
3. Aggregate revised mean count values to all stratifications of Naomi outputs.
4. Calculate calibrated mean for proportion indicators.
5. Adjust outputs for all statistics (mean, median, mode, se, range).
6. Aggregate totals spectrum\_calibration table.

The "logistic" scaling method converts fine counts to logit proportions, then uses numerical optimisation to solve the logit-scale adjustment to the fine district/sex/age proportions such that the adjusted proportions times the denominator sums to the target value.

Calibration proceeds sequentially through the following indicators.

- PLHIV
- Number of residents on ART
- Number unaware of HIV status
- Number of new infections
- Number of attending ANC by district

Calibration of a previous indicator may affect the denominator for the next indicator. This does not affect the calculation for proportional scaling, but will affect logistic scaling. Inconsistent selections for calibration levels or stratifications could result in a denominator smaller than a target numerator for a particular value. This will throw an error for logistic scaling methods.

The number of attending ARG clients is always calibrated proportionally by sex and five-year age group to the number of residents attending.

---

 cmc\_date

*Convert Date to Century Month Code (CMC)*


---

### Description

Converts a date to DHS Century Month Code (CMC).

### Usage

```
cmc_date(date)
```

### Arguments

date            a Date vector

### Details

CMC date is defined as the number of months since 1900:

$$cmc = (year - 1900) * 12 + month$$

**Value**

an integer vector of CMC dates

**References**

[https://dhsprogram.com/Data/Guide-to-DHS-Statistics/Organization\\_of\\_DHS\\_Data.htm?rhtocid=\\_4\\_2\\_0#Structure\\_of\\_DH1](https://dhsprogram.com/Data/Guide-to-DHS-Statistics/Organization_of_DHS_Data.htm?rhtocid=_4_2_0#Structure_of_DH1)

**Examples**

```
cmc_date(Sys.Date())  
cmc_date(as.Date("1987-02-11", format = "%Y-%m-%d"))
```

---

create_adj_matrix	<i>Create adjacency matrix from spatial polygons</i>
-------------------	--

---

**Description**

Create adjacency matrix from spatial polygons

**Usage**

```
create_adj_matrix(sh)
```

**Arguments**

sh                    a spatial polygons object

**Details**

Input sh can be either sf class or SpatialPolygons from sp. Uses spdep package to convert shapefile to neighbor list to adjacency matrix.

**Value**

an adjacency matrix with 1s and 0s

---

create_areas	<i>Create an Areas Object</i>
--------------	-------------------------------

---

**Description**

Constructs and validates an areas object as an S3 class.

**Usage**

```
create_areas(  
  levels = NULL,  
  hierarchy = NULL,  
  boundaries = NULL,  
  area_merged = NULL  
)
```

**Arguments**

levels	Data frame of area level metadata.
hierarchy	Data frame defining area hierarchy and area-level data.
boundaries	an sf object with boundary geometry for each area_id
area_merged	A merged version of levels, hierarchy, and boundaries.

**Value**

An object of class naomi\_areas

**Examples**

```
data(demo_area_levels)  
data(demo_area_hierarchy)  
data(demo_area_boundaries)  
  
areas <- create_areas(demo_area_levels, demo_area_hierarchy, demo_area_boundaries)  
areas
```

---

create_area_aggregation
-------------------------

---

*Long data frame mapping area hierarchy areas to model level areas*

---

**Description**

Long data frame mapping area hierarchy areas to model level areas

**Usage**

```
create_area_aggregation(model_area_ids, areas, drop_partial_areas = TRUE)
```

**Arguments**

model\_area\_ids vector of model areas.

areas naomi\_areas object.

drop\_partial\_areas

Drop areas from output if some children are missing (default TRUE).

**Examples**

```
area_merged <- read_area_merged(system.file("extdata/demo_areas.geojson", package = "naomi"))
areas <- create_areas(area_merged = area_merged)
model_area_ids <- area_merged$area_id[area_merged$area_level == 4]
```

```
create_area_aggregation(model_area_ids, areas)
```

---

create\_edge\_list      *Create edgelist from adjacency matrix*

---

**Description**

Create edgelist from adjacency matrix

**Usage**

```
create_edge_list(adj_matrix)
```

**Arguments**

adj\_matrix      Adjacency matrix

---

cut\_naomi\_age\_group      *Cut Five Year Age Groups*

---

**Description**

Wrapper for [cut()] to return five year age groups with

**Usage**

```
cut_naomi_age_group(age)
```

**Arguments**

age                    a vector of ages.

**Value**

a vector of strings with five year age groups.

**See Also**

get\_age\_groups

---

data\_type\_labels            *Get id to label mapping for calibration plot data type*

---

**Description**

Get id to label mapping for calibration plot data type

**Usage**

data\_type\_labels()

**Value**

List of ID & label mappings

---

demo\_anc\_testing            *Programme data: Malawi routine HIV service delivery data*

---

**Description**

Datasets containing Malawi National HIV programme quarterly reported data about:

- Reported number currently on ART.
- Number of women receiving antenatal care and final antenatal HIV testing.

**Usage**

demo\_anc\_testing

demo\_art\_number

**Format**

TODO: Link to demo\_areas dataset demo\_art\_number is a dataset with 2042 observations on the following 5 variables:

area\_id District area id – see demo\_areas dataset.  
sex Sex as a character vector – all values are "both".  
age\_group Value "Y015\_999" is age 15+, "Y000\_014" is age 0-14.  
calendar\_quarter Calendar quarter.  
art\_current Number currently on ART at the end of the quarter.  
art\_new Number newly initiated ART during the quarter.

TODO: Link to demo\_areas dataset demo\_anc\_testing is a dataset with 923 observations on the following 8 variables:

area\_id District area id – see demo\_areas dataset.  
quarter\_id Quarter ID (integer).  
anc\_clients Number of unique ANC clients during quarter.  
anc\_known\_pos Number of ANC clients already known to be HIV positive prior to first ANC visit and not tested for HIV.  
anc\_already\_art Number of ANC clients already on ART prior to first ANC visit.  
anc\_tested Number of ANC clients tested for HIV and received result.  
anc\_tested\_pos Number of ANC clients testing HIV positive.  
anc\_known\_pos Number of ANC clients not tested for HIV due to documented recent HIV negative status (optional).  
births\_facility Number of life births recorded at health facilities..

An object of class tbl\_df (inherits from tbl, data.frame) with 830 rows and 10 columns.

**Details**

Data are sourced from quarterly reporting spreadsheets produced via the Quarterly Integrated HIV Program Reports and publicly available from Malawi Ministry of Health: Department of HIV & AIDS website.

Data are provided by the Ministry of Health, Malawi with the following disclaimer:

1. The data in this file are owned by the Ministry of Health (MOH), Malawi.
2. These data are shared in order to support the National HIV Program.
3. Data in this file may not be used by Third Parties for further analysis and dissemination without prior written approval of the Director of the Department for HIV and AIDS, MOH.
4. Patient-level data are routinely collected by health facility staff using standard monitoring tools. This file contains facility-level aggregates that have been generated by facility staff and that were verified from primary records during quarterly National HIV Program Supervision (coordinated of the Department of HIV and AIDS). While every effort is made to ensure high data quality, individual records may not be complete and accurate.



5. The official interpretation of these data is presented in Quarterly Integrated HIV Program Reports and any divergent interpretation may be misleading and is not supported by the MOH. Analysis and interpretation of the data requires detailed understanding of the methods and constraints of the Malawi's National M&E System.

The number of ART by age 0-14 and age 15+ is not reported in aggregate quarterly reporting data. For model illustration purposes, this is approximated as 94% of all ART clients are age 15+ based on Spectrum model outputs which were validated against age distributions from electronic medical records by the Malawi HIV estimates team.

Dataset could contain one observation per district per quarter for all 32 districts for all quarters between Q3 2011 and Q3 2018, but the number of observations is fewer than  $32 * 29 = 928$  due to a few missing observations.

### Source

Department of HIV & AIDS, Ministry of Health, Malawi. <http://hiv.health.gov.mw/index.php/our-documents>

### See Also

[quarter\\_labels](#), [get\\_age\\_groups](#).

---

demo\_area\_boundaries *Malawi area boundaries.*

---

### Description

Malawi area boundaries.

### Usage

demo\_area\_boundaries

### Format

sf object representing area boundaries

### Source

Malawi

---

demo\_area\_hierarchy    *Malawi area hierarchy*

---

**Description**

Malawi area hierarchy

**Usage**

demo\_area\_hierarchy

**Format**

Data frame

**Source**

Malawi

---

demo\_area\_levels    *Malawi area levels.*

---

**Description**

Malawi area levels.

**Usage**

demo\_area\_levels

**Format**

Data frame

**Source**

Malawi

---

demo\_population\_agesex

*Malawi age-sex stratified population data*

---

**Description**

Malawi age-sex stratified population data

**Usage**

demo\_population\_agesex

**Format**

Data frame

**Source**

Malawi

---

demo\_survey\_biomarker *Malawi survey biomarkers*

---

**Description**

Malawi survey biomarkers

**Usage**

demo\_survey\_biomarker

**Format**

Data frame

**Source**

Malawi

---

demo\_survey\_clusters *Malawi survey clusters*

---

**Description**

Malawi survey clusters

**Usage**

demo\_survey\_clusters

**Format**

Data frame

**Source**

Malawi

---

demo\_survey\_hiv\_indicators  
*Malawi HIV indicators*

---

**Description**

Malawi HIV indicators

**Usage**

demo\_survey\_hiv\_indicators

**Format**

Data frame

**Source**

Malawi

---

demo\_survey\_individuals  
*Malawi survey individuals*

---

**Description**

Malawi survey individuals

**Usage**

demo\_survey\_individuals

**Format**

Data frame

**Source**

Malawi

---

demo\_survey\_meta      *Malawi survey metadata*

---

**Description**

Malawi survey metadata

**Usage**

demo\_survey\_meta

**Format**

Data frame

**Source**

Malawi

---

demo\_survey\_regions    *Malawi survey regions*

---

**Description**

Malawi survey regions

**Usage**

```
demo_survey_regions
```

**Format**

Data frame

**Source**

Malawi

---

disaggregate\_0to4\_outputs  
*Disaggregate age 0-4 outputs to <1 / 1-4*

---

**Description**

Disaggregate output indicators for the 0-4 age group proportional to Spectrum distribution for each age group.

**Usage**

```
disaggregate_0to4_outputs(output, naomi_mf)
```

**Arguments**

output	a naomi_output object
naomi_mf	a naomi_mf object

---

district\_barplot      *Create district bar plot*

---

### Description

Create district bar plot

### Usage

```
district_barplot(
  naomi_output,
  indicator,
  age,
  sex_disag,
  level = NULL,
  calendar_quarter = 2,
  label_format,
  x_title = NULL,
  fig_title = NULL,
  legend_title = NULL,
  colour_palette = "blue",
  above_label = t_("ABOVE"),
  below_label = t_("BELOW")
)
```

### Arguments

naomi_output	Naomi output object or path to naomi output zip folder
indicator	Indicator to plot
age	Age group to plot
sex_disag	Sexes to plot
level	Area level to plot, default all
calendar_quarter	Naomi projection quarter to filter data to, default is calendar_quarter_t2
label_format	Label format for y labels
x_title	Title for the figure x axis, blank by default
fig_title	Title for the figure, blank by default,
legend_title	Title for the legend
colour_palette	Names or hexcode value for colours to differentiate districts with values above and below the national average. Can be specified as a preset colour palette using "red", "green" and "blue" or manually as a named list: cols <- c(above_colour = "red", below_colour = "blue").
above_label	Label for districts with values higher than national average, default is "Above"
below_label	Label for districts with values higher than national average, default is "Below"

**Value**

District barplot

---

dropdown_buttons	<i>Automatically make drop drop buttons for plotly filter</i>
------------------	---

---

**Description**

Automatically make drop drop buttons for plotly filter

**Usage**

```
dropdown_buttons(df, var)
```

**Arguments**

df	Input data frame
var	Column

**Value**

Drop down buttons for plotly filter

---

extract_pjnz_naomi	<i>Extract Model Inputs from Spectrum PJNZ</i>
--------------------	--

---

**Description**

If the .shiny90 file does not exist within the .PJNZ, the function will silently not return values, even if `extract_shiny90 = TRUE`.

**Usage**

```
extract_pjnz_naomi(pjnz_list, extract_shiny90 = TRUE)
```

**Arguments**

pjnz_list	Vector of filepaths to Spectrum PJNZ file.
extract_shiny90	Logical; whether to attempt to extract .shiny90 zip

**Value**

A data.frame with Spectrum indicators.



**Examples**

```
pjnz <- system.file("extdata/demo_mwi2019.PJNZ", package = "naomi")
spec <- extract_pjnz_naomi(pjnz)
```

---

```
extract_pjnz_program_data
```

*Extract ART and ANC testing program data inputs from Spectrum PJNZ*

---

**Description**

Extract ART and ANC testing program data inputs from Spectrum PJNZ

**Usage**

```
extract_pjnz_program_data(pjnz_list)
```

**Arguments**

`pjnz_list`      Vector of filepaths to Spectrum PJNZ file.

**Value**

A list with a two data.frames of ANC testing data and number on ART, respectively.

**Examples**

```
pjnz <- system.file("extdata/demo_mwi2019.PJNZ", package = "naomi")
spec <- extract_pjnz_program_data(pjnz)
```

---

```
extract_shiny90_age_sex
```

*Export proportion aware by five year age group from Shiny90*

---

**Description**

Export estimates for proportion aware of status from a Shiny90 output file to five-year age groups 15-19 to 50+.

**Usage**

```
extract_shiny90_age_sex(shiny90_path, pjnz_path = NULL, years = NULL)
```

**Arguments**

shiny90\_path    file path to .shiny90 digest file.  
 pjnz\_path        file path to PJNZ file  
 years            year(s) to generate estimates; an integer or a vector of integers. If NULL, all years available in estimates are returned (default).

**Details**

The 'artnum' divided by 'plhiv' columns in the output will give a different ART coverage than Spectrum output for the same year because these outputs are based on the internal mid-year ART coverage in Spectrum, while Spectrum reports an end of year (Dec 31) ART coverage.

**Value**

A data frame consisting of the number of PLHIV, aware of status and on ART by sex and five year age group 15-49 and age 50+ for specified years.

**Examples**

```
pjnz <- system.file("extdata/demo_mwi2019.PJNZ", package = "naomi")
shiny90dir <- tempfile()
utils::unzip(pjnz, "malawi.zip.shiny90", exdir = shiny90dir)
shiny90_path <- file.path(shiny90dir, "malawi.zip.shiny90")

extract_shiny90_age_sex(shiny90_path, pjnz, year = 2010:2019)
```

---

 fit\_tmb

*Fit TMB model*


---

**Description**

Fit TMB model

**Usage**

```
fit_tmb(
  tmb_input,
  outer_verbose = TRUE,
  inner_verbose = FALSE,
  max_iter = 250,
  progress = NULL
)
```

**Arguments**

tmb_input	Model input data
outer_verbose	If TRUE print function and parameters every iteration
inner_verbose	If TRUE then disable tracing information from TMB
max_iter	maximum number of iterations
progress	Progress printer, if null no progress printed

**Value**

Fit model.

---

get_age_groups	<i>Data frame of age groups</i>
----------------	---------------------------------

---

**Description**

Return a data frame consisting of master age groups

**Usage**

```
get_age_groups()
```

**Value**

data frame

---

get_anc_metadata	<i>Metadata for ANC plot types used in the input time series plots. Mainly for internal use.</i>
------------------	--

---

**Description**

Metadata for ANC plot types used in the input time series plots. Mainly for internal use.

**Usage**

```
get_anc_metadata(anc)
```

**Arguments**

anc	Path to file containing ANC data or ANC data object
-----	---

**Value**

List of plot\_types and cols\_keep (columns that are present in the actual data out of the known list of columns)

---

get\_area\_collection    *Get a collection of areas*

---

### Description

Get a collection of areas defined by a level and nested within a collection of higher level area(s).

### Usage

```
get_area_collection(areas, level = NULL, area_scope = NULL)
```

### Arguments

areas	Data frame representing a valid area hierarchy.
level	The level to return area collection
area_scope	area_ids to subset return level, possible a vector.

### Details

This function recursively expands the area hierarchy to identify all areas in area\_level level which are nested within areas defined by area\_scope.

If level = NULL areas are returned at the lowest level of the hierarchy. An error is thrown if level is not reflected in the hierarchy.

If area\_scope = NULL all areas at area\_level level are returned. In this case the recursion is short-cut and simply filters the hierarchy to the desired level for efficiency, but it is equivalent to initialising the recursion at the highest level of the hierarchy.

If area\_scope is not null, a column area\_scope is returned indicating the area in which each returned area is nested. area\_scope can accept a vector of area\_ids and they do not have to be at the same level. If the level of an area in area\_scope is higher than area\_level level, nothing is returned. An error is thrown if any area\_scope are not recognized.

TODO: Should be an example - where is demo\_areas, demo\_area\_geom? data(demo\_areas, demo\_area\_geom)

```
areas <- get_area_collection(demo_areas, level = 3, area_scope = c("MWI.1", "MWI.3.5")) areas
%>% left_join(demo_area_geom %>% filter(type == "boundary")) %>% sf::st_as_sf() %>% gg-
plot() + geom_sf()
```

---

get_art_metadata	<i>Metadata for ART plot types used in the input time series plots. Mainly for internal use.</i>
------------------	--

---

**Description**

Metadata for ART plot types used in the input time series plots. Mainly for internal use.

**Usage**

```
get_art_metadata(art)
```

**Arguments**

anc                    Path to file containing ART data or ART data object

**Value**

List of plot\_types and cols\_keep (columns that are present in the actual data out of the known list of columns) and calendar\_quarters

---

get_five_year_age_groups	<i>Get 5 year age groups</i>
--------------------------	------------------------------

---

**Description**

Get 5 year age groups

**Usage**

```
get_five_year_age_groups()
```

**Value**

ID for 5 year age groups

**Examples**

```
get_five_year_age_groups()
```

---

get_metadata	<i>Get indicator metadata as a data frame</i>
--------------	---

---

**Description**

Get indicator metadata as a data frame

**Usage**

```
get_metadata()
```

**Value**

Metadata about indicators as a list

---

get_meta_indicator	<i>Get indicator metadata</i>
--------------------	-------------------------------

---

**Description**

Get indicator metadata

**Usage**

```
get_meta_indicator()
```

**Value**

data.frame of indicator ids, labels, descriptions, and parameter mapping.

**Examples**

```
get_meta_indicator()
```

---

get\_plotting\_metadata *Get plotting metadata for a particular country*

---

**Description**

Get plotting metadata for a particular country

**Usage**

```
get_plotting_metadata(iso3)
```

**Arguments**

iso3                    iso3 code of country to get metadata for or blank for default configuration.

**Value**

List of plotting metadata about how to locate data for a specific indicator, data type and plot type. Also returns metadata about colour scheme to use for that country and indicator.

**Examples**

```
get_plotting_metadata("MWI")
```

---

get\_plot\_type\_column\_metadata  
*Return the translated label & description for a set of plot types*

---

**Description**

Return the translated label & description for a set of plot types

**Usage**

```
get_plot_type_column_metadata(plot_type)
```

**Arguments**

plot\_type            Plot type ids

**Value**

For each plot type the label and description as a list of lists containing id, label and description

---

```
get_spec_aggr_interpolation
    Interpolate Spectrum to quarter_id
```

---

**Description**

Interpolate Spectrum to quarter\_id

**Usage**

```
get_spec_aggr_interpolation(spec_aggr, calendar_quarter_out)
```

**Arguments**

spec\_aggr            a data from of 5-year age group aggregate Spectrum estimates  
calendar\_quarter\_out        calendar quarter for desired output time point

---

```
hintr_calibrate        Calibrate hintr_output
```

---

**Description**

Take a previously generated hintr\_output object and calibrate. Format response as another hintr\_output object.

**Usage**

```
hintr_calibrate(  
  output,  
  calibration_options,  
  plot_data_path = tempfile(fileext = ".qs"),  
  calibrate_output_path = tempfile(fileext = ".qs")  
)
```

**Arguments**

output                A hintr\_output object.  
calibration\_options    A set of calibration options  
plot\_data\_path        Path to store calibrated output indicators as a qs.  
calibrate\_output\_path    Path to store data required for re-calibrating model.

**Value**

Calibrated hintr\_output object



---

hintr\_calibrate\_plot *Get data for hintr calibrate plot*

---

**Description**

Takes hintr\_output object, reads data and prepares data in format needed for plotting calibrate bar-chart to compare calibrated, spectrum and unadjusted estimates for a set of indicators.

**Usage**

```
hintr_calibrate_plot(output)
```

**Arguments**

output            A hintr\_output object

**Value**

Calibrated, unadjusted and spectrum estimates of indicators

---

hintr\_comparison\_plot *Get data for hintr comparison plot*

---

**Description**

Takes hintr\_output object, reads data and prepares data in format needed for plotting comparison barchart to compare input and output data.

**Usage**

```
hintr_comparison_plot(output)
```

**Arguments**

output            A hintr\_output object

**Value**

Calibrated, unadjusted and spectrum estimates of indicators

---

hintr\_prepare\_agyw\_download  
*Prepare AGYW tool download*

---

**Description**

Prepare AGYW tool download

**Usage**

```
hintr_prepare_agyw_download(output, pjnz, path = tempfile(fileext = ".xlsx"))
```

**Arguments**

<code>pjnz</code>	Path to input PJNZ file
<code>path</code>	Path to save output file
<code>hintr_output</code>	object

**Value**

Path to output file and metadata for file

---

hintr\_prepare\_coarse\_age\_group\_download  
*Prepare coarse age group download*

---

**Description**

Prepare coarse age group download

**Usage**

```
hintr_prepare_coarse_age_group_download(  
  output,  
  path = tempfile(fileext = ".zip")  
)
```

**Arguments**

<code>output</code>	hintr output object
<code>path</code>	Path to save output file

**Value**

Path to output file and metadata for file

---

hintr\_prepare\_comparison\_report\_download  
*Prepare comparison report download*

---

**Description**

Prepare comparison report download

**Usage**

```
hintr_prepare_comparison_report_download(  
    output,  
    path = tempfile(fileext = ".html")  
)
```

**Arguments**

path	Path to save output file
hintr_output	object

**Value**

Path to output file and metadata for file

---

hintr\_prepare\_spectrum\_download  
*Prepare spectrum download*

---

**Description**

Prepare spectrum download

**Usage**

```
hintr_prepare_spectrum_download(  
    output,  
    path = tempfile(fileext = ".zip"),  
    notes = NULL,  
    vmmc_file = NULL  
)
```

**Arguments**

output	hintr output object
path	Path to save output file
notes	Optional, user added notes from front end of app as a string
vmmc_file	Optional file object, with path, filename and hash for VMMC input

**Value**

Path to output file and metadata for file

---

hintr\_prepare\_summary\_report\_download  
*Prepare summary report download*

---

**Description**

Prepare summary report download

**Usage**

```
hintr_prepare_summary_report_download(  
  output,  
  path = tempfile(fileext = ".html")  
)
```

**Arguments**

output	hintr output object
path	Path to save output file

**Value**

Path to output file and metadata for file

---

hintr\_run\_model      *Run the model and save output*

---

**Description**

This prepares the model inputs from data and options and saves output as a qs file.

**Usage**

```
hintr_run_model(  
  data,  
  options,  
  model_output_path = tempfile(fileext = ".qs"),  
  validate = TRUE  
)
```

**Arguments**

data	List of paths to input data files.
options	List of model run options (see details).
model_output_path	Path to store model output as qs. Used in calibrating model and producing output downloads.
validate	If FALSE validation of inputs & data will be skipped.

**Details**

The data argument must be a list specifying paths to the following:

- pjnz
- shape
- population
- survey data
- anc\_testing (optional)
- art\_number (optional)

Each item in list can either be a character containing the path to the file or another list of the format:

```
list( path = "path/to/file", hash = "file_hash", filename = "file" )
```

The options argument must be a list specifying minimally:

- area\_scope
- area\_level
- calendar\_quarter\_t1
- calendar\_quarter\_t2
- calendar\_quarter\_t3
- survey\_prevalence
- survey\_art\_coverage
- survey\_recently\_infected
- include\_art\_t1
- include\_art\_t2
- anc\_prevalence\_year1
- anc\_prevalence\_year2
- anc\_art\_coverage\_year1
- anc\_art\_coverage\_year2

**Value**

Paths to output files

---

```
hintr_validate_programme_data
```

*Run validation for subnational programme data input*

---

### Description

This can throw validation errors or warnings which will be shown to user in naomi web app

### Usage

```
hintr_validate_programme_data(naomi_spectrum_comparison)
```

### Arguments

```
naomi_spectrum_comparison
```

Comparison table of aggregated subnational Naomi and national Spectrum programme data created by `prepare_art_spectrum_comparison()` or `prepare_anc_spectrum_comparison()`

---

Imports

*We need to import these as they are used by dependencies (first90) unalified and need to be available in the package environment for the tests to be able to run in a background progress (i.e. when running in parallel)*

---

### Description

We need to import these as they are used by dependencies (first90) unalified and need to be available in the package environment for the tests to be able to run in a background progress (i.e. when running in parallel)

---

```
interpolate_population_agesex
```

*Log-linear interpolation of age/sex stratified population*

---

### Description

Log-linear interpolation of age/sex stratified population

### Usage

```
interpolate_population_agesex(population_agesex, calendar_quarters)
```

**Arguments**

population\_agesex  
 a subset of the population\_agesex.

calendar\_quarters  
 vector of calendar quarters to return interpolation.

**Details**

zoo::na.approx() is used to interpolate log(population).

**Value**

A data.frame with same columns as pop\_agesex interpolated to times.

**See Also**

[convert\\_quarter\\_id\(\)](#)

**Examples**

```
## Interpolate Malawi population at level 2 (Zone) at two time points
data(demo_population_agesex)
calendar_quarters <- c("CY2016Q1", "CY2018Q3")
pop_interp <- interpolate_population_agesex(demo_population_agesex, calendar_quarters)
```

---

log_linear_interp	<i>Log-linear interpolation of NA values</i>
-------------------	--

---

**Description**

Log-linear interpolation of NA values

**Usage**

```
log_linear_interp(y, x, rule = 2, replace_na = 0)
```

**Arguments**

y  
 vector of output values, possibly with NAs

x  
 vector of points to interpolate (no NAs)

rule  
 rule for extrapolating outside range (see [approx\(\)](#))

replace\_na  
 value to replace if interpolation evaluates to NA

**Examples**

```
log_linear_interp(c(100, 105, NA, 110), 1:4)
log_linear_interp(c(NA, 105, NA, 110), 1:4)
log_linear_interp(c(NA, 105, NA, 110, NA), 1:5, rule = 1)
log_linear_interp(c(NA, 105, NA, 110, NA), 1:5, rule = 2)
log_linear_interp(c(NA, NA, 37), 1:3, rule = 2)
```

---

map\_outputs

*Create output map plot*


---

**Description**

Create output map plot

**Usage**

```
map_outputs(
  naomi_geom,
  indicator,
  calendar_quarter,
  age,
  sex_disag,
  level = NULL,
  colour_palette = "red",
  fig_title = NULL,
  legend_title = NULL,
  legend_label = ggplot2::waiver(),
  breaks = 6
)
```

**Arguments**

naomi_geom	Labelled naomi output object with geometry
indicator	The indicator to generate plot for
calendar_quarter	Naomi projection quarter to filter supplied data to.
age	The age group to include in plots
sex_disag	Sex diagggregation to use in plot,
level	Integer, corresponding to desired area level to filter supplied data to. Default is lowest area level available in supplied data.
colour_palette	Names or hexcode value for colour scale to be applied to to lowest and highest value in the supplied data. Can be specified as a preset colour palette using "red", "green" and "blue" or manually as a named list: cols = c( start_colour = "red", end_colour = "blue").



fig_title	Title for the figure, blank by default
legend_title	Title for the legend, blank by default
legend_label	Legend label
breaks	Number of break points to create in scale

**Value**

Mapped outputs

---

naomi_model_frame	<i>Construct Model Frames and Adjacency Structures</i>
-------------------	--

---

**Description**

Construct Model Frames and Adjacency Structures

**Usage**

```
naomi_model_frame(
  area_merged,
  population_agesex,
  spec,
  scope = area_merged$area_id[area_merged$area_level == min(area_merged$area_level)],
  level = max(area_merged$area_level),
  calendar_quarter1,
  calendar_quarter2,
  calendar_quarter3,
  calendar_quarter4 = "CY2024Q3",
  calendar_quarter5 = "CY2025Q3",
  age_groups = get_five_year_age_groups(),
  sexes = c("male", "female"),
  omega = 0.7,
  rita_param = list(OmegaT0 = 130/365, sigma_OmegaT = ((142 - 118)/365)/(2 *
    stats::qnorm(0.975)), betaT0 = 0, sigma_betaT = 1e-05, ritaT = 1),
  sigma_u_sd = 1,
  artattend = TRUE,
  artattend_t2 = FALSE,
  artattend_log_gamma_offset = -4,
  anchor_home_district = TRUE,
  rho_paed_15to49f_ratio = TRUE,
  rho_paed_x_term = FALSE,
  alpha_xst_term = FALSE,
  logit_nu_mean = 2,
  logit_nu_sd = 0.3,
  spectrum_population_calibration = "national",
  output_aware_plhiv = TRUE,
```

```

    adjust_area_growth = FALSE,
    psnu_level = NULL
)

```

### Arguments

area_merged	Merged version of area hierarchy
population_agesex	Population by age group and sex
spec	Spec
scope	The collection of area IDs to be modelled. Defaults to all area ids.
level	Admin level
calendar_quarter1	Calendar quarter at time 1 ("CYyyyyQq")
calendar_quarter2	Calendar quarter at time 2 ("CYyyyyQq")
calendar_quarter3	Calendar quarter at time 3 ("CYyyyyQq")
calendar_quarter4	Calendar quarter at time 4 ("CYyyyyQq")
calendar_quarter5	Calendar quarter at time 5 ("CYyyyyQq")
age_groups	Age groups to include in model frame
sexes	Sexes
omega	Omega
rita_param	rita_param
sigma_u_sd	sigma_u_sd
artattend	logical; whether to estimate neighboring district ART attendance
artattend_t2	logical; whether to allow time-varying neighboring district ART attendance
artattend_log_gamma_offset	logit offset for neighboring district ART attendance
anchor_home_district	logical; whether to include home district random in ART attendance specification
rho_paed_15to49f_ratio	logical; to model paediatric prevalence as ratio of 15-49 female prevalence
rho_paed_x_term	logical; to include area interaction for paediatric prevalence
alpha_xst_term	logical; to include district-sex-time interaction for ART coverage. Default FALSE.
logit_nu_mean	mean of logit viral load suppression.
logit_nu_sd	standard deviation of logit viral load suppression.
spectrum_population_calibration	character string values "national", "subnational", "none"

adjust_area_growth	TRUE/FALSE: adjust PLHIV population for net change in cohort size
psnu_level	Level at which to output PEPFAR Data Pack outputs for planning. This doesn't have anything to do with calibration, but this is a convenient place to allow user to edit this in model workflow. Default value NULL indicates to read PSNU level from internal database.

### Details

Argument `spectrum_population_calibration` determines whether to calibrate population inputs to match Spectrum population by age and sex. If the Spectrum file is a single national Spectrum file, then options "national" and "subnational" return the same results.

### Value

Naomi model frame

---

`naomi_objective_function_r`  
*R implementation of Naomi model*

---

### Description

R implementation of Naomi model

### Usage

```
naomi_objective_function_r(d, p)
```

### Arguments

<code>d</code>	list of data inputs (from <a href="#">prepare_tmb_inputs()</a> )
<code>p</code>	list of parameter values (from <a href="#">prepare_tmb_inputs()</a> )

### Details

This is primarily written to enable stepping through the TMB model line by line.

### Value

A list consisting of (1) reported objects from the TMB model, and (2) the value of the objective function.

---

naomi\_output\_frame      *Model Frame and Linear Transform for Aggregated Model Outputs*

---

### Description

Model Frame and Linear Transform for Aggregated Model Outputs

### Usage

```
naomi_output_frame(
  mf_model,
  area_aggregation,
  age_groups = unique(mf_model[["age_group"]]),
  sexes = unique(mf_model[["sex"]])
)
```

### Arguments

mf_model	Model frame
area_aggregation	data.frame with columns area_id and model_area_id.
age_groups	Age groups to include in aggregated outputs.
sexes	Sexes to include in aggregated outputs.

### Value

A list consisting of a data.frame mf and a sparse matrix A. The data frame mf provides a model matrix defining the stratifications for the outputs. The sparse matrix A defines a linear transform to aggregate the rows of mf\_model to the rows of the output mf

---

output\_naomi\_warning      *Generate naomi warning for specific strata of simulated outputs*

---

### Description

Generate naomi warning for specific strata of simulated outputs

### Usage

```
output_naomi_warning(naomi_output, ind, threshold, locations)
```

### Arguments

naomi_output	Naomi output object
ind	Naomi output indicator
threshold	Numerical threshold to trigger warning
locations	Location where warning should be displayed in front end

---

output_package	<i>Build output package from fit</i>
----------------	--------------------------------------

---

**Description**

Build output package from fit

**Usage**

```
output_package(naomi_fit, naomi_data, na.rm = FALSE)
```

**Arguments**

naomi_fit	Fitted naomi model
naomi_data	Naomi model frame with data
na.rm	Whether to remove NA values when calculating summary statistics, default FALSE

**Details**

The argument `na.rm = TRUE` allows the output package to be produced when there are errors due to missing values when generating outputs. This is only for debugging purposes to review results when there are errors. NA values in simulated model results typically mean poor model fit or non-convergence that needs to be addressed.

**Value**

List containing output indicators and metadata.

---

pop_pyramid_outputs	<i>Create population pyramid plot</i>
---------------------	---------------------------------------

---

**Description**

Create population pyramid plot

**Usage**

```
pop_pyramid_outputs(
  naomi_output,
  calendar_quarter = 2,
  indicator,
  colour_palette = "blue",
  x_title = NULL,
  y_title = t_("AGE_GROUP"),
```

```

    fig_title = NULL,
    legend_label = abs,
    masc_label = t_("SEX_MALE"),
    fem_label = t_("SEX_FEMALE")
  )

```

### Arguments

naomi_output	Naomi output object or path to naomi output zip folder
calendar_quarter	Naomi projection quarter to filter data to, default is calendar_quarter_t2
colour_palette	Names or hexcode value for right and left side of population pyramid. Can be specified as a preset colour palette using "red", "green" and "blue" or manually as a named list: cols <- c( left_clour = "red", right_colour = "blue").
x_title	Title for the figure x axis, blank by default
y_title	Title for the figure y axis, default is "Age Group"
fig_title	Title for the figure, blank by default,
masc_label	Label for male sex group, default is "Male"
fem_label	Label for female sex group, default is "Female"

### Value

Pop pyramid

---

```
prepare_input_time_series_anc
```

*Prepare data for ANC input time series plots*

---

### Description

Take uploaded ANC and shape file paths or objects and format as data which can be used to draw input time series graphs.

### Usage

```
prepare_input_time_series_anc(anc, shape)
```

### Arguments

anc	Path to file containing ANC data or ANC data object
shape	Path to file containing geojson areas data or shape sf object

### Value

Data formatted for plotting input ANC time series containing columns area\_id, area\_name, area\_level, area\_level\_label, parent\_area\_id, area\_sort\_order, time\_period, year, quarter, calendar\_quarter, area\_hierarchy, plot, value and missing\_ids

---

prepare\_input\_time\_series\_art  
*Prepare data for ART input time series plots*

---

**Description**

Take uploaded ART and shape file paths and format as data which can be used to draw input time series graphs.

**Usage**

```
prepare_input_time_series_art(art, shape)
```

**Arguments**

art	Path to file containing ART data or ART data object
shape	Path to file containing geojson areas data or area data object

**Value**

Data formatted for plotting input time series containing columns area\_id, area\_name, area\_level, area\_level\_label, parent\_area\_id, area\_sort\_order, time\_period, year, quarter, calendar\_quarter, area\_hierarchy, plot, value and missing\_ids

---

prepare\_spectrum\_naomi\_comparison  
*Compare aggregated subnational Naomi + spectrum totals for comparison table*

---

**Description**

Compare aggregated subnational Naomi + spectrum totals for comparison table

**Usage**

```
prepare_spectrum_naomi_comparison(art, anc, shape, pjnz)
```

**Arguments**

art	Path to file containing ART data or ART data object
anc	Path to file containing ART data or ART data object
shape	Path to file containing geojson areas data or areas data object
pjnz	Path to zip file containing spectrum pjnz file/s

---

```
prepare_tmb_inputs      Prepare inputs for TMB model.
```

---

**Description**

Prepare inputs for TMB model.

**Usage**

```
prepare_tmb_inputs(naomi_data, report_likelihood = 1L)
```

**Arguments**

`naomi_data`      Naomi data object

`report_likelihood`  
                   Option to report likelihood in fit object (default true).

`anchor_home_district`  
                   Option to include random effect home district attractiveness to retain residents on ART within home districts (default true).

**Value**

Inputs ready for TMB model

**See Also**

[select\\_naomi\\_data](#)

---

```
quarter_year_labels    Time period indexing
```

---

**Description**

Time periods are indexed by integers for efficiency and precision. Quarters are indexed as the number of quarters since the beginning of 1900:  $\$quarter\_id = (year - 1900) * 4 + quarter\$$ .

**Usage**

```
quarter_year_labels(quarter_id)

quarter_number(quarter_id)

quarter_labels(quarter_id)

calendar_quarter_labels(calendar_quarter)
```



```
calendar_quarter_labels_short(calendar_quarter)
year_labels(quarter_id)
convert_quarter_id(year, quarter)
convert_calendar_quarter(year, quarter)
calendar_quarter_to_quarter_id(calendar_quarter)
quarter_id_to_calendar_quarter(quarter_id)
calendar_quarter_to_year(calendar_quarter)
calendar_quarter_to_quarter(calendar_quarter)
```

### Arguments

quarter_id	vector of integer quarter IDs.
calendar_quarter	Vector of calendar quarters to convert.
year	vector of integer years.
quarter	vector of integer quarters (1,2,3,4).

### Details

Quarters are labelled as "Jan-Mar", "Apr-Jun", "Jul-Sep", "Oct-Dec" instead of "Q1", "Q2", "Q3", "Q4" to avoid confusion between calendar quarters and offset fiscal year quarters.

### Examples

```
quarter_ids <- convert_quarter_id(c(2009, 2017), c(3, 1))
quarter_ids
calender_quarters <- convert_calendar_quarter(c(2009, 2017), c(3, 1))
quarter_number(quarter_ids)
quarter_labels(quarter_ids)
year_labels(quarter_ids)
quarter_year_labels(quarter_ids)
calendar_quarter_labels("CY2015Q2")
calendar_quarter_to_year("CY2015Q2")
calendar_quarter_to_quarter("CY2015Q2")
```

---

read_hintr_output	<i>Read hintr output</i>
-------------------	--------------------------

---

**Description**

Read the hintr model output or plot data saved as a qs or an rds file. This is the data saved from `hintr_run_model` or `hintr_calibrate` before the output zip is generated. This uses the file extension to identify the reading function to use.

**Usage**

```
read_hintr_output(path)
```

**Arguments**

path	Path to the file
------	------------------

**Value**

The read data

---

read_population	<i>Read Naomi structured input files</i>
-----------------	--

---

**Description**

Read Naomi structured input files

**Usage**

```
read_population(file)
read_survey_indicators(file)
read_art_number(file, all_columns = FALSE)
read_anc_testing(file)
read_area_merged(file)
```

**Arguments**

file	A path to a file.
------	-------------------

**Examples**

```
anc_path <- system.file("extdata/demo_anc_testing.csv", package = "naomi")
anc_testing <- read_anc_testing(anc_path)
```

---

`read_spectrum_projection_name`

*Read Spectrum Projection Name from Spectrum PJNZ*

---

**Description**

Read Spectrum Projection Name from Spectrum PJNZ

**Usage**

```
read_spectrum_projection_name(pjnz)
```

**Arguments**

`pjnz` file path to Spectrum PJNZ file.

**Value**

Spectrum projection name as character string.

**Examples**

```
pjnz <- system.file("extdata/demo_mwi2019.PJNZ", package = "naomi")
read_spectrum_projection_name(pjnz)
```

---

`read_spectrum_region_code`

*Read Subnational Region Code from Spectrum PJNZ*

---

**Description**

Read Subnational Region Code from Spectrum PJNZ

**Usage**

```
read_spectrum_region_code(pjnz)
```

**Arguments**

`pjnz` file path to Spectrum PJNZ file.

**Details**

The region code is 0 if a national Spectrum file.

**Value**

Spectrum subnational region code as an integer.

**Examples**

```
pjnz <- system.file("extdata/demo_mwi2019.PJNZ", package = "naomi")
read_spectrum_region_code(pjnz)
```

---

read\_spectrum\_region\_name

*Read Subnational Region Name from Spectrum PJNZ*

---

**Description**

Read Subnational Region Name from Spectrum PJNZ

**Usage**

```
read_spectrum_region_name(pjnz)
```

**Arguments**

pjnz                    file path to Spectrum PJNZ file.

**Details**

Value NA corresponds to region code 0 for a national Spectrum file.

**Value**

Spectrum subnational region name as a string. Returns NA if no subnational region.

**Examples**

```
pjnz <- system.file("extdata/demo_mwi2019.PJNZ", package = "naomi")
read_spectrum_region_name(pjnz)
```

---

report_tmb	<i>Calculate Posterior Mean and Uncertainty Via TMB</i> sdreport()
------------	--

---

**Description**

Calculate Posterior Mean and Uncertainty Via TMB sdreport()

**Usage**

```
report_tmb(naomi_fit)
```

**Arguments**

naomi_fit	Fitted TMB model.
-----------	-------------------

---

sample_tmb	<i>Sample TMB fit</i>
------------	-----------------------

---

**Description**

Sample TMB fit

**Usage**

```
sample_tmb(
  fit,
  nsample = 1000,
  rng_seed = NULL,
  random_only = TRUE,
  verbose = FALSE
)
```

**Arguments**

fit	The TMB fit
nsample	Number of samples
rng_seed	seed passed to set.seed.
random_only	Random only
verbose	If TRUE prints additional information.

**Value**

Sampled fit.

---

save_output	<i>Save outputs to zip file</i>
-------------	---------------------------------

---

### Description

Save outputs to zip file

### Usage

```
save_output(
  filename,
  dir,
  naomi_output,
  notes = NULL,
  vmmc_path = NULL,
  overwrite = FALSE,
  with_labels = FALSE,
  boundary_format = "geojson",
  single_csv = FALSE,
  export_datapack = !single_csv
)
```

### Arguments

filename	Name of file to create
dir	Directory to create zip in
naomi_output	Naomi output object
notes	Notes to include in output zip
vmmc_path	Path to VMMC excel workbook
overwrite	If TRUE overwrite any existing file
with_labels	If TRUE save indicator ids with labels
boundary_format	Either geojson or shp for saving boundary as geojson or shape format
single_csv	If TRUE only output the csv of indicators, otherwise save the metadata too
export_datapack	If TRUE save CSV of PEPFAR datapack indicators.
options	Naomi model options

### Value

Path to created zip file

---

save\_output\_package    *Save outputs to zip file*

---

**Description**

Save outputs to zip file

**Usage**

```
save_output_package(
  naomi_output,
  filename,
  dir,
  notes = NULL,
  overwrite = FALSE,
  with_labels = FALSE,
  boundary_format = "geojson",
  single_csv = FALSE,
  export_datapack = !single_csv
)
```

```
read_output_package(path)
```

**Arguments**

naomi_output	Naomi output object
filename	Name of file to create
dir	Directory to create zip in
notes	Notes to include in output zip
overwrite	If TRUE overwrite any existing file
with_labels	If TRUE save indicator ids with labels
boundary_format	Either geojson or shp for saving boundary as geojson or shape format
single_csv	If TRUE only output the csv of indicators, otherwise save the metadata too
export_datapack	If TRUE save CSV of PEPFAR datapack indicators.
path	Path to output zip file.

**Value**

Path to created zip file

---

scale\_gmrf\_precision    *Scale of GMRF precision matrix*

---

### Description

This function scales the precision matrix of a GMRF such that the geometric mean of the marginal variance is one.

### Usage

```
scale_gmrf_precision(
  Q,
  A = matrix(1, ncol = ncol(Q)),
  eps = sqrt(.Machine$double.eps)
)
```

### Arguments

Q	Precision matrix for a GMRF.
A	Linear constraint for Q.
eps	Value of the small constant added to the diagonal of Q for invertibility.

### Details

This implements the same thing as `INLA::inla.scale.model`. The marginal variance of each connected component is one.

---

scatter\_plotly    *Plotly scatterplot data inputs and naomi outputs*

---

### Description

Plotly scatterplot data inputs and naomi outputs

### Usage

```
scatter_plotly(
  df,
  ind,
  quarter,
  input_data,
  input_data_type,
  age_disag = "Y015_049",
  sex_disag = "both"
)
```



**Arguments**

df	Inputs_outputs dataframe containing matched model estimates and data inputs.
ind	Indicator filter.
quarter	Calendar quarter filter.
input_data	Input data name.
input_data_type	Input data type.
age_disag	Age group filter.
sex_disag	Sex filter.

---

select_naomi_data	<i>Select data for model fitting</i>
-------------------	--------------------------------------

---

**Description**

Select data for model fitting

**Usage**

```
select_naomi_data(
  naomi_mf,
  survey_hiv_indicators,
  anc_testing,
  art_number,
  prev_survey_ids,
  artcov_survey_ids,
  recent_survey_ids,
  vls_survey_ids = NULL,
  artnum_calendar_quarter_t1 = naomi_mf[["calendar_quarter1"]],
  artnum_calendar_quarter_t2 = naomi_mf[["calendar_quarter2"]],
  anc_clients_year_t2 =
  year_labels(calendar_quarter_to_quarter_id(naomi_mf[["calendar_quarter2"]])),
  anc_clients_year_t2_num_months = 12,
  anc_prev_year_t1 =
  year_labels(calendar_quarter_to_quarter_id(naomi_mf[["calendar_quarter1"]])),
  anc_prev_year_t2 =
  year_labels(calendar_quarter_to_quarter_id(naomi_mf[["calendar_quarter2"]])),
  anc_artcov_year_t1 = anc_prev_year_t1,
  anc_artcov_year_t2 = anc_prev_year_t2,
  use_kish_prev = TRUE,
  deff_prev = 1,
  use_kish_artcov = TRUE,
  deff_artcov = 1,
  use_kish_recent = TRUE,
  deff_recent = 1,
```

```

    use_kish_vls = TRUE,
    deff_vls = 1,
    use_survey_aggregate = FALSE,
    spec_program_data = NULL
  )

```

## Arguments

naomi_mf	A Naomi model frame object.
survey_hiv_indicators	Data frame of survey estimates, or NULL to exclude any survey data.
anc_testing	Data frame of ANC routine testing outcomes, or NULL to exclude any ANC data.
art_number	Data frame of number currently receiving ART, or NULL to exclude any ART data.
prev_survey_ids	A character vector of survey_ids for prevalence data.
artcov_survey_ids	A character vector of survey_ids for ART coverage data.
recent_survey_ids	A character vector of survey_ids for recent HIV infection status.
vls_survey_ids	A character vector of survey_ids for survey VLS among all HIV+ persons.
artnum_calendar_quarter_t1	Calendar quarter for first time point for number on ART.
artnum_calendar_quarter_t2	Calendar quarter for second time point for number on ART.
anc_clients_year_t2	Calendar year (possibly multiple) for number of ANC clients at year 2.
anc_prev_year_t1	Calendar year (possibly multiple) for first time point for ANC prevalence.
anc_prev_year_t2	Calendar year (possibly multiple) for second time point for ANC prevalence.
anc_artcov_year_t1	Calendar year (possibly multiple) for first time point for ANC ART coverage.
anc_artcov_year_t2	Calendar year (possibly multiple) for second time point for ANC ART coverage.
deff_prev	Approximate design effect for survey prevalence.
deff_artcov	Approximate design effect for survey ART coverage.
deff_recent	Approximate design effect for survey proportion recently infected.
deff_vls	Approximate design effect for survey viral load suppression.
spec_program_data	Object of class "spec_program_data" consisting of aggregated program data inputs to Spectrum. Provided for checking against Naomi inputs. If NULL then checks are not conducted.
anc_clients_year_t2_num_monhts	Number of months of reporting reflected in the year(s) recorded in anc_clients_year_t2.

## Details

See example datasets for examples of required template for data sets. \*\_survey\_ids must be reflected in survey\_hiv\_indicators.

ART coverage and VLS survey data should not be included from the same survey. This is checked by the function call and will throw an error.

The deff\_\* arguments are approximate design effects used to scale the effective sample size for survey observations. Stratified design effects are will not be the same as full survey DEFF and there is not a straightforward way to approximate these.

The option use\_aggregate\_survey = TRUE allows for aggregate versions of survey data to be used in model fitting, for example age 15-49 prevalence instead of five-year age group prevalence or province-level survey data instead of district level data. This maybe useful if cluster coordinates or survey microdata are not available. This option assumes that the survey\_hiv\_indicators is already subsetted to exactly the data to be used. All stratifications must also appear in the naomi\_data\$mf\_out stratifications.

## See Also

[demo\\_survey\\_hiv\\_indicators](#), [demo\\_anc\\_testing](#), [demo\\_art\\_number](#), [convert\\_quarter\\_id](#)

---

spread\_areas

*Spread area hierarchy to wide format*

---

## Description

Spread area hierarchy to wide format

## Usage

```
spread_areas(
  areas,
  min_level = min(areas$area_level),
  max_level = max(areas$area_level)
)
```

## Arguments

areas            area hierarchy data.frame

min\_level        integer specifying the minimum level

max\_level        integer specifying the maximum level

TODO: Make this an example - where is areas.rds? areas <- readRDS(system.file("extdata/areas/areas.rds", package = "naomi")) areas\_wide <- spread\_areas(areas)

---

subset\_naomi\_output     *Subset the results returned in Naomi output package*

---

### Description

Subset the results returned in Naomi output package

### Usage

```
subset_naomi_output(  
  naomi_output,  
  area_id = NULL,  
  area_level = NULL,  
  sex = NULL,  
  age_group = NULL,  
  calendar_quarter = NULL,  
  indicator = NULL,  
  drop = FALSE,  
  check_list = TRUE  
)
```

### Arguments

naomi_output	Naomi output object.
area_id	vector of area_ids to include/exclude.
area_level	vector of area_levels to include/exclude.
sex	vector of sexes to include/exclude.
age_group	vector of age_groups to include/exclude.
calendar_quarter	vector of calendar_quarters to include/exclude.
indicator	vector of indicators to include/exclude.
drop	logical whether to drop the supplied indices instead of keep only the supplied indices (default).
check_list	logical whether to check that supplied values are in the output package to be subsetted.

### Details

If arguemnts are NULL (default), no subsetting is done on that dimension.

By default the argument `check_list = TRUE` means an error will be thrown if any of the values in the vectors to subset are not found in the `naomi_output` object supplied. This might be set to `FALSE` for some batch processing applications, for example of the `naomi_output` could have already been partially subsetted.

**Value**

A naomi output package with a subset of results.

---

subset\_output\_package *Resave a subsetted Naomi output package*

---

**Description**

This function reads an output package, subsets it using [subset\\_naomi\\_output\(\)](#) and resaves the output package.

**Usage**

```
subset_output_package(path, output_path, ...)
```

**Arguments**

path	file path to naomi output package.
output_path	path to resave subsetted output package.
...	arguments to <a href="#">subset_naomi_output()</a> .

**Details**

See [?subset\\_naomi\\_output\(\)](#) for subsetting arguments and options.

**Value**

path to saved output package.

**See Also**

[subset\\_naomi\\_output\(\)](#)

---

survey_mf	<i>Prepare model frames for survey datasets</i>
-----------	---

---

**Description**

Prepare model frames for survey datasets

**Usage**

```
survey_mf(
  survey_ids,
  indicator,
  survey_hiv_indicators,
  naomi_mf,
  use_kish = TRUE,
  deff = 1,
  min_age = 0,
  max_age = 80,
  use_aggregate = FALSE
)
```

**Arguments**

survey_ids	Survey IDs
indicator	Indicator to filter, character string
survey_hiv_indicators	Survey HIV indicators
naomi_mf	Naomi model frame
use_kish	Logical whether to use Kish effective sample size
deff	Assumed design effect for scaling effective sample size
min_age	Min age for calculating recent infection
max_age	Max age for calculating recent infection
use_aggregate	Logical; use aggregate survey data as provided instead of subsetting fine area/sex/age group stratification.

---

th_map	<i>ggplot2 theme for plotting maps</i>
--------	--

---

**Description**

ggplot2 theme for plotting maps

**Usage**

```
th_map()
```

---

 validate\_model\_options

*Validate a set of model options*


---

### Description

This validates that a set of model options can be used to run the model

### Usage

```
validate_model_options(data, options)
```

### Arguments

data	The set of input data for the model run
options	Key-value list of model options

### Value

TRUE if valid otherwise throw an error

---

 write\_datapack\_csv

*Export naomi outputs to PEPFAR Data Pack format*


---

### Description

Export naomi outputs to PEPFAR Data Pack format

### Usage

```
write_datapack_csv(naomi_output, path, psnu_level = NULL, dmppt2_output = NULL)
```

### Arguments

naomi_output	a naomi_output object.
path	path to save Data Pack CSV.
psnu_level	area_level for PEPFAR PSNU to export. If NULL, first looks in lookup table for the correct area_level, and if not defaults to the highest level of the area hierarchy.
dmppt2_output	data frame containing the <i>Datapack inputs</i> sheet of DMPPT2 output file.

**Details**

The results will include Naomi outputs for area\_id at the specified level, irrespective of whether a Datim map\_id is available.

If a datim ID is available, the Datim map\_name will be used in the column psnu. Otherwise the Naomi area\_name will be used for the column psnu.

PEPFAR indicator codes for Data Pack are in this Datim view: <https://www.datim.org/api/sqlViews/DotdxKrNZxG/data.html>

PEPFAR PSNU UIDs are in this Datim view: <https://www.datim.org/api/sqlViews/gsaaxFM8ZN0/data.html+css>

(Replace the extensions .html+css with .csv to download tables as CSV.)

---

write\_navigator\_checklist

*Write UNAIDS Estimates Navigator checklist CSV*

---

**Description**

Write UNAIDS Estimates Navigator checklist CSV

**Usage**

```
write_navigator_checklist(naomi_output, path)
```

**Arguments**

naomi_output	a naomi_output object.
path	path to save Navigator checklist CSV.



# Index

## \* datasets

- demo\_anc\_testing, 15
  - demo\_area\_boundaries, 17
  - demo\_area\_hierarchy, 18
  - demo\_area\_levels, 18
  - demo\_population\_agesex, 19
  - demo\_survey\_biomarker, 19
  - demo\_survey\_clusters, 20
  - demo\_survey\_hiv\_indicators, 20
  - demo\_survey\_individuals, 21
  - demo\_survey\_meta, 21
  - demo\_survey\_regions, 22
- 
- add\_output\_labels, 4
  - age\_bar\_plotly, 4
  - aggregate\_anc, 5
  - aggregate\_art, 5
  - align\_inputs\_outputs, 6
  - anc\_testing\_artcov\_mf
    - (anc\_testing\_prev\_mf), 6
  - anc\_testing\_clients\_mf
    - (anc\_testing\_prev\_mf), 6
  - anc\_testing\_prev\_mf, 6
  - approx(), 39
  - artnum\_mf, 7
  - assert\_pjnz\_shiny90, 7
- 
- bar\_plotly, 8
  - build\_hierarchy\_label, 8
- 
- calculate\_prevalence\_art\_coverage, 9
  - calendar\_quarter\_labels
    - (quarter\_year\_labels), 48
  - calendar\_quarter\_labels\_short
    - (quarter\_year\_labels), 48
  - calendar\_quarter\_to\_quarter
    - (quarter\_year\_labels), 48
  - calendar\_quarter\_to\_quarter\_id
    - (quarter\_year\_labels), 48
  - calendar\_quarter\_to\_year
    - (quarter\_year\_labels), 48
  - calibrate\_outputs, 9
  - cmc\_date, 11
  - convert\_calendar\_quarter
    - (quarter\_year\_labels), 48
  - convert\_quarter\_id, 59
  - convert\_quarter\_id
    - (quarter\_year\_labels), 48
  - convert\_quarter\_id(), 39
  - create\_adj\_matrix, 12
  - create\_area\_aggregation, 13
  - create\_areas, 13
  - create\_edge\_list, 14
  - cut\_naomi\_age\_group, 14
- 
- data\_type\_labels, 15
  - demo\_anc\_testing, 15, 59
  - demo\_area\_boundaries, 17
  - demo\_area\_hierarchy, 18
  - demo\_area\_levels, 18
  - demo\_art\_number, 59
  - demo\_art\_number (demo\_anc\_testing), 15
  - demo\_population\_agesex, 19
  - demo\_survey\_biomarker, 19
  - demo\_survey\_clusters, 20
  - demo\_survey\_hiv\_indicators, 20, 59
  - demo\_survey\_individuals, 21
  - demo\_survey\_meta, 21
  - demo\_survey\_regions, 22
  - disaggregate\_0to4\_outputs, 22
  - district\_barplot, 23
  - dropdown\_buttons, 24
- 
- extract\_pjnz\_naomi, 24
  - extract\_pjnz\_program\_data, 25
  - extract\_shiny90\_age\_sex, 25
- 
- fit\_tmb, 26
  - get\_age\_groups, 17, 27

- get\_anc\_metadata, 27
- get\_area\_collection, 28
- get\_art\_metadata, 29
- get\_five\_year\_age\_groups, 29
- get\_meta\_indicator, 30
- get\_metadata, 30
- get\_plot\_type\_column\_metadata, 31
- get\_plotting\_metadata, 31
- get\_spec\_aggr\_interpolation, 32
  
- hintr\_calibrate, 32
- hintr\_calibrate\_plot, 33
- hintr\_comparison\_plot, 33
- hintr\_prepare\_agyw\_download, 34
- hintr\_prepare\_coarse\_age\_group\_download, 34
- hintr\_prepare\_comparison\_report\_download, 35
- hintr\_prepare\_spectrum\_download, 35
- hintr\_prepare\_summary\_report\_download, 36
- hintr\_run\_model, 36
- hintr\_validate\_programme\_data, 38
  
- Imports, 38
- interpolate\_population\_agesex, 38
  
- log\_linear\_interp, 39
  
- map\_outputs, 40
  
- naomi\_model\_frame, 41
- naomi\_objective\_function\_r, 43
- naomi\_output\_frame, 44
  
- output\_naomi\_warning, 44
- output\_package, 45
- output\_package(), 10
  
- pop\_pyramid\_outputs, 45
- prepare\_input\_time\_series\_anc, 46
- prepare\_input\_time\_series\_art, 47
- prepare\_spectrum\_naomi\_comparison, 47
- prepare\_tmb\_inputs, 48
- prepare\_tmb\_inputs(), 43
  
- quarter\_id\_to\_calendar\_quarter (quarter\_year\_labels), 48
- quarter\_labels, 17
- quarter\_labels (quarter\_year\_labels), 48
  
- quarter\_number (quarter\_year\_labels), 48
- quarter\_year\_labels, 48
  
- read\_anc\_testing (read\_population), 50
- read\_area\_merged (read\_population), 50
- read\_art\_number (read\_population), 50
- read\_hintr\_output, 50
- read\_output\_package (save\_output\_package), 55
- read\_population, 50
- read\_spectrum\_projection\_name, 51
- read\_spectrum\_region\_code, 51
- read\_spectrum\_region\_name, 52
- read\_survey\_indicators (read\_population), 50
- report\_tmb, 53
  
- sample\_tmb, 53
- save\_output, 54
- save\_output\_package, 55
- scale\_gmrf\_precision, 56
- scatter\_plotly, 56
- select\_naomi\_data, 48, 57
- spread\_areas, 59
- subset\_naomi\_output, 60
- subset\_naomi\_output(), 61
- subset\_output\_package, 61
- survey\_mf, 62
  
- th\_map, 62
  
- validate\_model\_options, 63
  
- write\_datapack\_csv, 63
- write\_navigator\_checklist, 64
  
- year\_labels (quarter\_year\_labels), 48