Package: LBoM.tools (via r-universe)

October 24, 2024

Type Package

Title Tools for Curating London Bills of Mortality and Registrar General Data

Version 0.1.0

Maintainer Steve Walker <swalk@mcmaster.ca>

Description Part of an open toolchain for processing infectious disease datasets available through the IIDDA data repository.

License GPL (>= 3)

Depends R (>= 3.5.0), iidda

Imports dplyr, lubridate, purrr, stringr, tidyr, unpivotr, glue, readr, tidyxl

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3.9000

Repository https://canmod.r-universe.dev

RemoteUrl https://github.com/canmod/LBoM.tools

RemoteRef HEAD

RemoteSha f8c74a8df7426af398d74eb6efdf4949cb803397

Contents

`assert_current_scope` *Assert Current Scope*

Description

Scope of the data.

Usage

```
assert_current_scope(data_table, metadata)
```

Arguments

`data_table`	table containing data from tidyxl::xlsx_cells output
`metadata`	result of `get_tracking_metadata`

Value

filtered data list after entities that are out of scope have been removed. Data is in tidyxl::xlsx_cells format where each record corresponds to one cell in source data.

See Also

Other scope: `empty_cols()`, `empty_rows()`, `exclude_fields()`, `header_rows()`, `inconsistent_column_names()`, `missing_fields()`, `question_marks()`

combine_sheets *Combine Sheets*

Description

Combine Sheets

Usage

```
combine_sheets(data_table, time_metadata, dataset_type)
```

Arguments

data_table	Data frame returned by ‘read_digitized_data‘.
time_metadata	List returned by make_time_metadata.
dataset_type	Type of dataset (e.g., Mortality, Birth)

convert_to_date *Convert to date*

Description

Convert date string to date data type.

Usage

```
convert_to_date(y, m, d, period)

convert_vec_to_date(y, m, d, period)
```

Arguments

y	string containing 4-digit year
m	string containing month numbers (1-12) or ‘NA‘.
d	string containing day numbers (1-31) or ‘NA‘.
period	a character string indicating whether the years are the start or end of a period. This argument is only available when ‘m‘ and ‘d‘ are ‘NA‘. Valid inputs are "start"(January 1) or "end"(December 31), default is NULL.

Value

Date object either a valid date or NA_Date_

Functions

- convert_vec_to_date(): Vectorized version of date conversion

create_date_fields *Create date fields*

Description

Creates two new fields of class 'Date' named 'period_start_date' and 'period_end_date' in input table using existing fields containing date range information.

Usage

```
create_date_fields(data_table)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output

Value

'data_table' containing additional fields named 'period_start_date' and 'period_end_date'

date_range_issues *Date range issues*

Description

Identify gaps and overlaps in weekly date ranges in Excel source data sheets.

Usage

```
date_range_issues(data_table)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output

Value

all records in 'data_table' that correspond to gaps and overlaps in weekly date range data

See Also

Other data_quality_issues: `invalid_dates()`, `repeated_field_names()`, `unclassified_field_names()`

`empty_cols` *Empty cols*

Description

Identify entire columns in Excel sheets that contain no numeric or text data. These columns are outside the boundaries of the data table contained in the sheet but they may contain Excel formatting that causes the data reading functionality to perceive these columns as containing data.

Usage

```
empty_cols(data_table)
```

Arguments

`data_table`	table containing data from tidyxl::xlsx_cells output

Value

all records in ‘data_table‘ that correspond to entire Excel sheet columns with no data

See Also

Other scope: `assert_current_scope()`, `empty_rows()`, `exclude_fields()`, `header_rows()`, `inconsistent_column_names()`, `missing_fields()`, `question_marks()`

`empty_rows` *Empty rows*

Description

Identify entire rows in Excel sheets that contain no numeric or text data. These rows are outside the boundaries of the data table contained in the sheet but they may contain Excel formatting that causes the data reading functionality to perceive these rows as containing data.

Usage

```
empty_rows(data_table)
```

Arguments

`data_table`	table containing data from tidyxl::xlsx_cells output

Value

all records in ‘data_table‘ that correspond to entire Excel sheet rows with no data

See Also

Other scope: assert_current_scope(), empty_cols(), exclude_fields(), header_rows(), inconsistent_column_names(), missing_fields(), question_marks()

`exclude_fields` *Exclude fields*

Description

Temporarily exclude some fields that require additional work/thought on how they should be incorporated in the tidy data set. These include: - some parish total fields - parish grouping fields are now included in scope for the mortality data set using reference tables that map parish groups with individual parishes. Parish grouping fields for birth data is currently not in scope, however it can be included in scope by following the framework used for the mortality data set. - relative fields - fields that contain relative counts (ex. "increases in burials this week")

Usage

```
exclude_fields(data_table)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output

Value

all records in ‘data_table‘ that correspond to excluded fields

See Also

Other scope: assert_current_scope(), empty_cols(), empty_rows(), header_rows(), inconsistent_column_names(
missing_fields(), question_marks()

`filter_out_data_quality`
Data quality filter

Description

Create data quality issue report from input table with the option to filter out data quality issues.

Usage

```
filter_out_data_quality(data_table, metadata, filter_data = TRUE, skip = TRUE)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output
metadata	result of get_tracking_metadata
filter_data	boolean flag to filter out data quality issues. For the 'repeated_field_name' data quality issue, records are always filtered to create unique records despite this flag to ensure later pipeline processing steps can function.
skip	completely skip data quality checking and just return the input dataset

Value

Returns input 'data_table' with possibly additional unique records corresponding to the 'repeated_field_name' data quality issue when 'filter_data=FALSE'. If 'filter_data=TRUE' function will return 'data_table' after data quality issue records have been removed.

---

fix_tidy_data_columns *Fix Tidy Data Columns*

Description

Fix Tidy Data Columns

Usage

```
fix_tidy_data_columns(data_set)
```

Arguments

data_set	processed output from one of the data processing functions (e.g. process_mortality)

---

header_rows *Header rows*

Description

Identify entire rows in Excel sheets that contain header comments in at least one of the cells. These rows appear at the top of the Excel sheets and are assumed to start with a hash tag character.

Usage

```
header_rows(data_table)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output

Value

all records in ‘data_table‘ that correspond to header rows

See Also

Other scope: assert_current_scope(), empty_cols(), empty_rows(), exclude_fields(), inconsistent_column_nam
missing_fields(), question_marks()

`inconsistent_column_names` *Inconsistent time column names*

Description

Identify sheets that contain time data formatted differently than all other sheets. (One Excel file contains weekly time data with no ".from" or ".to" suffix for "year", "month", and "day" columns.)

Usage

```
inconsistent_column_names(data_table)
```

Arguments

data_table	table containing mortality data from tidyxl::xlsx_cells output

Value

all records in ‘data_table‘ that correspond to inconsistent time column names

See Also

Other scope: assert_current_scope(), empty_cols(), empty_rows(), exclude_fields(), header_rows(), missing_fields(), question_marks()

`invalid_dates` *Invalid dates*

Description

Identifies data records that correspond to invalid dates in Excel source data sheets. Invalid dates refer to all dates that are ‘NA_Date_‘ after ‘create_date_fields()‘ was executed. For example, non-existent dates (i.e.‘2000-02-30‘) are invalid dates.

Usage

```
invalid_dates(data_table)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output

Value

all records that correspond to invalid dates in ‘data_table‘.

See Also

Other data_quality_issues: date_range_issues(), repeated_field_names(), unclassified_field_names()

---

lbom_pre_processing *LBoM Preprocessing*

Description

LBoM Preprocessing

Usage

```
lbom_pre_processing(data, metadata)
```

Arguments

data	Dataset to be preprocessed in IIDDA pipelines.
metadata	Metadata in form returned by iidda::get_dataset_metadata

---

make_date_string *Concatenate Year, Month, and Day Fields into a yyyy-mm-dd String.*

Description

Concatenate Year, Month, and Day Fields into a yyyy-mm-dd String.

Usage

```
make_date_string(y, m, d, period = NULL)
```

Arguments

y	vector containing years
m	vector containing month numbers (1-12) or ‘NA‘.
d	vector containing day-of-month numbers (1-31) or ‘NA‘.
period	a character string indicating whether the years are the start or end of a period. This argument is only available when ‘m‘ and ‘d‘ are ‘NA‘. Valid inputs are "start"(January 1) or "end"(December 31), default is NULL.

make_date_string_vec *Make Date String*

Description

Concatenate Year, Month, and Day Fields into a yyyy-mm-dd String.

Usage

```
make_date_string_vec(y, m, d, period = NULL)

make_date_string_vec_year(y, m, d, period = NULL)
```

Arguments

y	vector containing years
m	vector containing month numbers (1-12) or 'NA'.
d	vector containing day-of-month numbers (1-31) or 'NA'.
period	a character string indicating whether the years are the start or end of a period. This argument is only available when 'm' and 'd' are 'NA'. Valid inputs are "start"(January 1) or "end"(December 31), default is NULL.

Functions

- `make_date_string_vec_year()`: Make date string from year information only

make_report_path *Make Report Path*

Description

Make the path to be used for storing reports and supporting output, by modifying the path to the tidy data to be found in the associated metadata.

Usage

```
make_report_path(metadata)
```

Arguments

metadata	result of `get_tracking_metadata`

make_time_metadata *Make Time Metadata*

Description

Make Time Metadata

Usage

```
make_time_metadata(data_table)
```

Arguments

data_table	Data frame returned by ‘read_digitized_data‘.

missing_fields *Missing fields*

Description

Identify columns in Excel sheets that contain numeric data with no associated field name. Given there is no field name the tidy data splitting functionality incorrectly assigns this data.

Usage

```
missing_fields(data_table)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output

Value

all records in ‘data_table‘ that correspond to entire Excel sheet columns with no field name

See Also

Other scope: `assert_current_scope()`, `empty_cols()`, `empty_rows()`, `exclude_fields()`, `header_rows()`, `inconsistent_column_names()`, `question_marks()`

process *Process Data*

Description

Process all data to convert from tidyxl::xlsx_cells format to long format. The output of this function contains data provenance information that allow one to trace each record back to a particular cell in the digitized Excel data.

Usage

```
process_population(data_table, metadata)

process_all_cause(data_table, metadata)

process_births(data_table, metadata)

process_plague(data_table, metadata)

process_mortality(data_table, metadata)
```

Arguments

data_table	table containing data in tidyxl::xlsx_cells format
metadata	result of get_tracking_metadata

Value

table containing counts in long format (by sex if applicable) and compact time metadata format, age data extracted from cause field names (if applicable)

Functions

- process_population(): Process population data
- process_all_cause(): Process all-cause mortality data
- process_births(): Process birth data
- process_plague(): Process plague data
- process_mortality(): Process mortality data

process_sheet *Process Excel Sheet*

Description

Process excel sheet to convert from tidyxl::xlsx_cells format to tidy long format

Usage

```
process_sheet(
  sheet,
  col_time_metadata,
  address_time_metadata,
  numeric_time_metadata,
 data_type = c("mortality", "all-cause-mortality", "births", "plague", "population")
)
```

Arguments

sheet	table containing one Excel sheet in tidyxl::xlsx_cells format
col_time_metadata	time metadata fields with "col_" prefix
address_time_metadata	time metadata fields with "address_" prefix
numeric_time_metadata	time metadata fields with "numeric_" prefix
data_type	TODO: describe data type

Value

processed sheet as a tibble

question_marks *Question marks*

Description

Identify cells that contain question marks. It is not clear what type of missing data is being represented by a question mark.

Usage

```
question_marks(data_table)
```

Arguments

data_table	table containing mortality data from tidyxl::xlsx_cells output

Value

all records in ‘data_table‘ that correspond to cells with question marks

See Also

Other scope: assert_current_scope(), empty_cols(), empty_rows(), exclude_fields(), header_rows(), inconsistent_column_names(), missing_fields()

---

read_source_RDS *Read Source RDS*

Description

Read in source RDS data

Usage

```
read_source_RDS(
  source_data_folder,
  data_category = list("mortality", "acm", "birth", "plague", "population",
    "unknown_data_category")
)
```

Arguments

source_data_folder	top level folder where source RDS files are saved
data_category	data category

Value

tibble containing all data in specified data category

repeated_field_names *Repeated field names*

Description

Field names that appear multiple times in Excel source data sheets. In order to have one data point per each time period (row in the sheet), the decision was made (https://github.com/davidearn/data_work/issues/192) to select the maximum numeric value for each record for these fields. Additionally, for each time period if both numeric and non-numeric values exist we prioritize numeric values, and in the presence of numeric ties we select only one of the numeric values. All records identified by these decisions are updated to have a new column number (that does not contain data in the digitized files) so they can easily be identified, and original columns are excluded in the final data set.

Usage

```
repeated_field_names(data_table)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output

Value

all records in ‘data_table‘ that correspond to repeated field names

See Also

Other data_quality_issues: date_range_issues(), invalid_dates(), unclassified_field_names()

split_source_excel *Split Source Excel*

Description

Read in source .xlsx data files and perform some data pre-processing. Splits data into data categories and saves resulting table as RDS file.

Usage

```
split_source_excel(
  source_data_folder,
  intermediate_data_folder,
 data_type = c("mortality", "all-cause-mortality", "births", "plague", "population")
)
```

Arguments

source_data_folder	folder where source data is saved.
intermediate_data_folder	top level folder for output, resulting RDS output file will be saved in intermediate_data_folder > data_type location.
data_type	data type to save as RDS file. Valid inputs are mortality, all-cause-mortality, births, plague, population. The default is `mortality`.

Value

the combined data frame that is saved in an RDS file

---

unclassified_field_names *Unclassified field names*

Description

Identify field names in Excel source data sheets that match regular expression patterns in representative_categories.csv that have been labelled data quality issues. Additionally, find field names that do not have a regular expression match, or have more that one match.

Usage

```
unclassified_field_names(data_table, representative_test_categories)
```

Arguments

data_table	table containing data from tidyxl::xlsx_cells output
representative_test_categories	data frame containing the reference table

Value

all records in ‘data_table‘ that correspond to unclassified field names in tidyxl::xlsx_cells format

See Also

Other data_quality_issues: `date_range_issues()`, `invalid_dates()`, `repeated_field_names()`

Index