

Package: tidyged.utils (via r-universe)

September 12, 2024

Title Utilities to Manage GEDCOM Files Using Tidyverse Principles

Version 0.4.3.9000

Description Various utilities to manage and clean family tree GEDCOM files using tidy dataframes.

URL <https://github.com/jl5000/tidyged.utils>

License GPL (>= 2)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE, roclets = c("` namespace", "` rd", "` roxytest::testthat_roclet"))

RoxygenNote 7.2.0

Imports dplyr, tibble, purrr, tidyr, lubridate, stringr, tidyged, tidyged.internals

Suggests testthat (>= 3.0.0), roxytest, rmarkdown, knitr

Remotes jl5000/tidyged, jl5000/tidyged.internals

Config/testthat/edition 3

VignetteBuilder knitr

Repository <https://jl5000.r-universe.dev>

RemoteUrl <https://github.com/jl5000/tidyged.utils>

RemoteRef HEAD

RemoteSha e395fee98c1203daed40e6d14372bca3451148a8

Contents

| | |
|-----------------------------|---|
| add_ancestors | 2 |
| age_now | 3 |
| arrange_records | 3 |
| consolidate_notes | 4 |
| date_diff | 4 |
| guess_age | 5 |

| | |
|--|----|
| guess_age_from_famg_events | 5 |
| guess_age_from_indi_events | 6 |
| identify_unused_records | 6 |
| insert_explicit_death_subrecords | 7 |
| insert_explicit_marr_types_all | 7 |
| make_xrefs_unique | 8 |
| merge_gedcoms | 8 |
| merge_records | 9 |
| migrate_records | 9 |
| order_famg_children_all | 10 |
| potential_duplicates | 10 |
| remove_change_dates | 11 |
| remove_duplicate_subrecords | 11 |
| remove_living | 12 |
| split_gedcom | 12 |

Index 14

| | |
|---------------|---|
| add_ancestors | <i>Add ancestor records for an individual</i> |
|---------------|---|

Description

This function adds placeholder Individual records for ancestors going back a specific number of generations.

Usage

```
add_ancestors(tg, xref, num_gen, inc_sex = TRUE)
```

Arguments

| | |
|---------|--|
| tg | A tidyged object. |
| xref | The xref of an Individual record to add ancestors for. |
| num_gen | The number of generations to create ancestors for. |
| inc_sex | Whether to populate the sex of the ancestors. This will ensure that there is one male and one female parent. Otherwise the sex will be assigned as "U" (undetermined). |

Details

This function may also create Family Group records and will not modify existing ancestors.

Value

A tidyged object with additional ancestor records.

| | |
|---------|--|
| age_now | <i>Determine the age of an individual now given their age on a previous date</i> |
|---------|--|

Description

Determine the age of an individual now given their age on a previous date

Usage

```
age_now(date_of_fact, age_at_fact, minimise = TRUE)
```

Arguments

| | |
|--------------|--|
| date_of_fact | A date string from the tidyged object. |
| age_at_fact | An age at event string from the tidyged object. |
| minimise | If date ranges or periods are used in the date, whether to choose the bounds which assume the minimum age. If this is FALSE, the maximum age is assumed. |

Value

A numeric value giving the current age in years. A numeric value less than zero means no determination could be made.

| | |
|-----------------|--|
| arrange_records | <i>Arrange all records in a tidyged object</i> |
|-----------------|--|

Description

This function groups together all records of a particular type and puts them in a specific order. This rearrangement makes no functional difference to the file, it just makes it more organised.

Usage

```
arrange_records(tg, order = "IFMSRN")
```

Arguments

| | |
|-------|--|
| tg | A tidyged object. |
| order | A character string indicating the desired order of records. The letters indicate (I)ndividual, (F)amily Group, (M)ultimedia, (S)ource, (R)epository, (N)ote. |

Value

An arranged tidyged object.

| | |
|-------------------|-------------------------------------|
| consolidate_notes | <i>Consolidate duplicated notes</i> |
|-------------------|-------------------------------------|

Description

Consolidate duplicated notes

Usage

```
consolidate_notes(tg, min_occurrences = 2)
```

Arguments

| | |
|-----------------|---|
| tg | A tidyged object. |
| min_occurrences | How many duplicates to prompt creating a new Note record. |

Value

A tidyged object with all notes consolidated.

| | |
|-----------|--|
| date_diff | <i>Determine the number of years between two dates</i> |
|-----------|--|

Description

Determine the number of years between two dates

Usage

```
date_diff(date1, date2 = tidyged::date_current(), minimise = TRUE)
```

Arguments

| | |
|----------|---|
| date1 | A date string from the tidyged object. |
| date2 | A date string from the tidyged object. If no date is given, today's date is used. |
| minimise | If date ranges or periods are used in the dates, whether to choose the bounds which assume the minimum date difference. If this is FALSE, the maximum date difference is assumed. |

Details

Doesn't yet handle dual years or BCE dates.

Value

A numeric value giving the number of years. A numeric value less than zero means no determination could be made.

| | |
|-----------|----------------------------------|
| guess_age | <i>Guess an individual's age</i> |
|-----------|----------------------------------|

Description

This function calculates an age for an individual based on their individual or family facts. It calculates age based on the date of the fact and their age when the fact applied.

Usage

```
guess_age(tg, xref, agg_fn = mean)
```

Arguments

| | |
|--------|---|
| tg | A tidyged object. |
| xref | The xref of an individual. |
| agg_fn | If multiple ages are calculated, the function to use to aggregate them. |

Value

A numeric value giving the estimated age. A numeric value less than zero means no determination could be made.

| | |
|----------------------------|---|
| guess_age_from_famg_events | <i>Guess an individual's age from their family group events</i> |
|----------------------------|---|

Description

This function takes an individual's family group events and calculates an estimated age based on the date of the event and their age when the event occurred.

Usage

```
guess_age_from_famg_events(tg, xref, agg_fn = mean)
```

Arguments

| | |
|--------|---|
| tg | A tidyged object. |
| xref | The xref of an individual. |
| agg_fn | If multiple ages are calculated, the function to use to aggregate them. |

Value

A numeric value giving the estimated age. A numeric value less than zero means no determination could be made.

guess_age_from_indi_events

Guess an individual's age from their facts

Description

This function takes an individual's attributes and events and calculates an estimated age based on the date of the fact and their age when the fact applied.

Usage

```
guess_age_from_indi_events(tg, xref, agg_fn = mean)
```

Arguments

| | |
|--------|---|
| tg | A tidyged object. |
| xref | The xref of an individual. |
| agg_fn | If multiple ages are calculated, the function to use to aggregate them. |

Value

A numeric value giving the estimated age. A numeric value less than zero means no determination could be made.

identify_unused_records

Identify unreferenced records

Description

This function identifies records that are not referenced in any other records.

Usage

```
identify_unused_records(tg)
```

Arguments

| | |
|----|-------------------|
| tg | A tidyged object. |
|----|-------------------|

Details

You would expect every record to be referenced by another in some way. For example, Individual records should reference Family Group records (and vice-versa), Repository records should be referenced by Source records, and Source records should be cited by other records.

You can use the output of this function with `tidyged::remove_records()` to remove them, or `tidyged::describe_records()` to find out more about them.

Value

A vector of xrefs that are not referenced anywhere else in the tidyged object.

```
insert_explicit_death_subrecords
```

Insert explicit death subrecords

Description

This function inserts explicit death subrecords for individuals who have a date of birth that makes them older than a maximum age.

Usage

```
insert_explicit_death_subrecords(  
  tg,  
  max_age = 120,  
  guess = FALSE,  
  explan_note = "This death event has been inferred automatically from other facts"  
)
```

Arguments

| | |
|-------------|---|
| tg | A tidyged object. |
| max_age | The maximum age to assume for a living person. |
| guess | If a date of birth cannot be found, whether to guess it from other information. |
| explan_note | Text to include in an explanatory note for any added death subrecords. An empty string will not add a note. |

Value

An updated tidyged object with additional death subrecords.

```
insert_explicit_marr_types_all
```

Insert explicit marriage subrecords

Description

This function inserts explicit marriage subrecords for Family Group records that do not have one.

Usage

```
insert_explicit_marr_types_all(tg)
```

Arguments

tg A tidyged object.

Details

The GEDCOM specification recommends that all marriage events have an explicit TYPE subrecord subordinate to the MARR relationship event. If one is not given, marriage is assumed.

Value

An updated tidyged object with additional marriage subrecords.

| | |
|-------------------|---|
| make_xrefs_unique | <i>Update xrefs in a tidyged object to make them unique from another tidyged object</i> |
|-------------------|---|

Description

This function ensures that all record xrefs across two tidyged objects are unique.

Usage

```
make_xrefs_unique(tg1, tg2)
```

Arguments

tg1 The first tidyged object.
 tg2 The second tidyged object. This will be the one that will be updated.

Value

The second tidyged object, where all xrefs have been made unique from the first object.

| | |
|---------------|----------------------------------|
| merge_gedcoms | <i>Merge two tidyged objects</i> |
|---------------|----------------------------------|

Description

Merge two tidyged objects

Usage

```
merge_gedcoms(tg1, tg2)
```


Arguments

tg1 The first tidyged object to merge.
 tg2 The second tidyged object to merge.

Value

A new tidyged object containing the records of both input objects. It will also have the same header and submitter information as the first input tidyged object.

| | |
|---------------|--|
| merge_records | <i>Combine multiple records into a single record</i> |
|---------------|--|

Description

This function takes multiple records and replaces them with a single record containing all of their subrecords. It does not remove duplicate subrecords.

Usage

```
merge_records(tg, xrefs)
```

Arguments

tg A tidyged object.
 xrefs The xrefs of the records to merge.

Value

A new tidyged object where all specified records have been merged.

| | |
|-----------------|--|
| migrate_records | <i>Copy all records from one tidyged object to another</i> |
|-----------------|--|

Description

This function takes two tidyged objects and copies all of the records in the second to the first, while making their identifiers unique (even if they may be duplicates).

Usage

```
migrate_records(tg1, tg2)
```

Arguments

- tg1 The first tidyged object and the one you want to use for header and submitter information.
- tg2 The second tidyged object.

Value

A tidyged object which contains the same header and submitter information as the first object and all records contained within both input objects.

order_famg_children_all

Order children in all Family Group records by birth date

Description

Order children in all Family Group records by birth date

Usage

```
order_famg_children_all(tg)
```

Arguments

- tg A tidyged object.

Value

The same tidyged object with rearranged children rows in the Family Group records.

potential_duplicates *Identify potentially duplicate records*

Description

Identify potentially duplicate records

Usage

```
potential_duplicates(tg)
```

Arguments

- tg A tidyged object.

Value

The same tidyged object, potentially with some records merged.

remove_change_dates *Remove all CHANge dates from a tidyged object*

Description

Remove all CHANge dates from a tidyged object

Usage

```
remove_change_dates(tg)
```

Arguments

tg A tidyged object.

Value

A tidyged object with all CHAN structures removed.

remove_duplicate_subrecords
 Remove duplicate subrecords from a tidyged record

Description

This function removes duplicate level 1 subrecords within a single record.

Usage

```
remove_duplicate_subrecords(tg, xref)
```

Arguments

tg A tidyged object.
xref The xref of the record to act on.

Value

The same tidyged object with duplicate subrecords removed from the specified record.

| | |
|---------------|---|
| remove_living | <i>Remove data for living individuals in a tidyged object</i> |
|---------------|---|

Description

Remove data for living individuals in a tidyged object

Usage

```
remove_living(
  tg,
  max_age = 120,
  guess = FALSE,
  remove_record = FALSE,
  explan_note = "Information on this individual has been redacted",
  remove_supp_records = TRUE
)
```

Arguments

| | |
|---------------------|---|
| tg | A tidyged object. |
| max_age | The maximum age to assume for a living person (if a date of birth is given). |
| guess | Whether to guess the age of individuals if no death event or date of birth is given and possibly retain them, or be cautious and remove them anyway (the default). |
| remove_record | Whether to remove the Individual records, or retain them as placeholders. |
| explan_note | Text to include in an explanatory note for any redacted records. An empty string will not add a note. |
| remove_supp_records | Whether to also remove supporting records (sources, repositories, notes, multimedia) associated with the living individuals. These may contain names and dates so it is probably best to remove them. |

Value

A tidyged object cleansed of information on living individuals.

| | |
|--------------|--|
| split_gedcom | <i>Split a tidyged object into two</i> |
|--------------|--|

Description

Split a tidyged object into two

Usage

```
split_gedcom(tg, xrefs)
```

Arguments

| | |
|-------|---|
| tg | A tidyged object to split. |
| xrefs | A vector of xrefs to put into the new tidyged object. |

Value

A new tidyged object containing the xrefs specified. It will also have the same header and submitter information as the input tidyged object.

Index

[add_ancestors](#), [2](#)
[age_now](#), [3](#)
[arrange_records](#), [3](#)

[consolidate_notes](#), [4](#)

[date_diff](#), [4](#)

[guess_age](#), [5](#)
[guess_age_from_famg_events](#), [5](#)
[guess_age_from_indi_events](#), [6](#)

[identify_unused_records](#), [6](#)
[insert_explicit_death_subrecords](#), [7](#)
[insert_explicit_marr_types_all](#), [7](#)

[make_xrefs_unique](#), [8](#)
[merge_gedcoms](#), [8](#)
[merge_records](#), [9](#)
[migrate_records](#), [9](#)

[order_famg_children_all](#), [10](#)

[potential_duplicates](#), [10](#)

[remove_change_dates](#), [11](#)
[remove_duplicate_subrecords](#), [11](#)
[remove_living](#), [12](#)

[split_gedcom](#), [12](#)