Package: parcel (via r-universe)

September 10, 2024

Title Convert real-world street addresses to county parcel identifiers

Version 0.11.1

Description Functions in parcel include cleaning, parsing, and creating shortened 'address stubs' to match real-world addresses to county-provided addresses with known parcel identifiers.

Config/reticulate list(packages = list(list(package = ``usaddress", version = ``0.5.10", pip = TRUE), list(package = ``csvdedupe", version = ``0.1.20", pip = TRUE), list(package = ``dedupe", version = ``2.0.23", pip = TRUE), list(package = ``numpy", version = ``1.25.2", pip = TRUE), list(package = ``dedupe-variable-address", version = ``0.0.8", pip = TRUE), list(package = ``scikit-learn", version = ``1.3.0", pip = TRUE)))

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports cincy (>= 1.0.2), dplyr, fr (>= 0.4.0), fs, purrr, reticulate (>= 1.31), rlang, stringr, tibble, tidyr

Suggests glue, mappp, readr, rvest, testthat (>= 3.0.0), xml2

Remotes geomarker-io/cincy, cole-brokamp/fr

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData true

URL http://geomarker.io/parcel/

Repository https://geomarker-io.r-universe.dev

RemoteUrl https://github.com/geomarker-io/parcel

RemoteRef HEAD

RemoteSha 97733e6085e87a6c06599f2d7f5499fad911669b

Contents

clean_address 2
create_address_stub 2
get_parcel_data 3
link_apt 4
link_parcel 4
tag_address 5

Index **6**

`clean_address` *clean address text*

Description

convert to lowercase, remove non-alphanumeric characters and excess whitespace (adapted from degauss-org/dht::clean_address)

Usage

```
clean_address(.x)
```

Arguments

`.x`	a vector of address character strings

Value

a vector of cleaned addresses

`create_address_stub` *extract the street number and name (i.e., "address stub") from address text*

Description

Input addresses are tagged into components and the `street_number` and `street_name` components are pasted together to create the address stub. If either the street_number or street_name are missing then the address_stub will be returned as missing. If `filter_zip` is TRUE, then addresses without a parsed 5-digit ZIP code in Hamilton County will have a missing address stub.

Usage

```
create_address_stub(.x, filter_zip = TRUE, ...)
```

Arguments

`.x` a vector of address character strings

`filter_zip` force addresses with non-Hamilton ZIP codes to have a missing address_stub?

`...` further arguments passed onto `tag_address()` (e.g., `clean`) (i.e., `cincy::zcta_tigris_2020$zcta_202`

Value

a vector of cleaned address stubs (street_number + street_name)

`get_parcel_data` *return parcel data for input addresses*

Description

This helper function produces a tibble of parcel data for an input vector of addresses. The `link_parcel()` function returns all possible matches above the `threshold` for each input address and this function chooses the single best match based on the maximum score. Note that one address can be linked to more than one parcel with the same match score (e.g., "323 Fifth" on https://wedge3.hcauditor.org/search_results). In this case, a special identifier, `TIED_MATCHES` is returned instead of a missing `parcel_id`. Addresses are subsequently tried to be matched with a known apartment complex using `link_apt()`. (Matched apartment complex psuedo-identifers take precedence over matched parcel identifers.) The `hamilton_online_parcels` tabular data resource is also linked based on `parcel_id`. For finer control of selecting matched parcels based on scores, use `link_parcel()` and `link_apt()`

Usage

```
get_parcel_data(x)
```

Arguments

`x` a vector of address character strings

Value

a tibble with the `input_addresses` defined in `x` in the first column, and columns corresponding to matched parcel characteristics from CAGIS and Auditor Online Summary website

`link_apt` *Link one address to parcel pseudo-identifiers for apartment complexes*

Description

To match a parcel to an apartment complex pseudo-identifier, it must contain:

- a Hamilton County ZIP code
- a street name matching the street names in `parcel:::apt_defs`
- a street number within the ranges for each pseudo-identifier in `parcel:::apt_defs`

Usage

```
link_apt(x)
```

Arguments

`x`	a single address character string

Value

apt pseudo-identifier character string; `NA` if not matched

`link_parcel` *link addresses to CAGIS parcel identifiers*

Description

This function uses the trained dedupe model included with the package to link one or more parcel identifiers to a vector of input addresses.

Usage

```
link_parcel(x, threshold = 0.2)
```

Arguments

`x`	a vector of address character strings
`threshold`	potential matches will only be returned if their `score` exceeds this value (from 0 to 1)

Details

Note that one address can be linked to more than one parcel (e.g., "323 Fifth" on https://wedge3.hcauditor.org/search_results). In this case, the input address will have multiple rows, one for each of the multiple matches.

Value

a tibble with a column of *unique*, matched addresses input as `x` along with columns for their `parcel_id`(s) and matching `score`(s) (use this as a lookup table for assigning parcel_id in other workflows, making decisions about what to do with multiple matches and matching thresholds, etc.)

`tag_address` *tag components of an address*

Description

This function relies on `usaddress` python library https://usaddress.readthedocs.io/en/latest/ It can be installed to a python virtual environment specific to R with: `py_install("usaddress", pip = TRUE)` (See the README for more details on installing and managing non-system installations of python with reticulate.

Usage

```
tag_address(address, clean = TRUE)
```

Arguments

`address`	a character string that is a United States mailing address
`clean`	clean addresses with `clean_address()` prior to tagging?

Details

This function uses a custom tag mapping to combine address components into the columns in the returned tibble (see https://usaddress.readthedocs.io/en/latest/#details for full definition of components):

- `street_number`: `AddressNumber`, `AddressNumberPrefix`, `AddressNumberSuffix`
- `street_name`: `StreetName`, `StreetNamePreDirectional`, `StreetNamePostDirectional`, `StreetNamePostModifier`, `StreetNamePostType`
- `city`: `PlaceName`
- `state`: `StateName`
- `zip`: the **first five characters** of `ZipCode`

If an address is not classified as a `Street Address` (i.e. `Intersection`, `PO Box`, or `Ambiguous`), then the columns in the returned component tibble will all be missing.

Value

a tibble with `street_number`, `street_name`, `city`, `state`, and `zip_code` columns

Index

clean_address, 2
create_address_stub, 2

get_parcel_data, 3

link_apt, 4
link_parcel, 4

tag_address, 5