Wikidata for data journalism (with R and ‘tidywikidatar’)

Dataharvest 2022

Giorgio Comai

(OBCT/CCI - EDJNet)

22 May 2022

Two big questions

1. how to you get Wikidata to throw at you the data you need

2. how to retrieve Wikidata's data in a way that won't give you headaches

Access Wikidata

Web front-end, nice to read: https://www.wikidata.org/wiki/Q162022

SPARQL:
https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/queries/examples

API version https://www.wikidata.org/w/api.php?action=wbgetentities&ids=Q162022

- machine readable
- nested lists, structure changes depending on values

What's the matter (for R users)?

- R users mostly hate nested lists
- some probably also hate SPARQL, but even more simply don't know much about SPARQL
- if you don't know what to expect, it's a pain to process data
- existing tools are not fit for the iterative data analysis process that is at the core of data journalism
- re-running an analysis with minor changes is a very common part of the workflow... without built-in caching, this can be painfully slow

`tidywikidata`r

Check out website with documentation and examples:
https://edjnet.github.io/tidywikidatar/

CRAN 0.5.2 downloads 406/month downloads 5783

- everything in tabular format
- one row, one piece of information
- easy local caching
- easy integration with `dplyr` piped routines
- get image credits from WikiMedia commons
- include Wikipedia in the exploration, or use it as a starting point

`tidywikidatar`

The basics

Enable local caching

```r
library(dplyr, warn.conflicts = FALSE)
library("tidywikidatar")

tw_enable_cache()
tw_set_cache_folder(
  path = fs::path(fs::path_home_r(),
                  "R",
                  "tw_data"))
tw_set_language(language = "en")
tw_create_cache_folder(ask = FALSE)
```

Or e.g. MySQL

```r
library(dplyr, warn.conflicts = FALSE)
library("tidywikidatar")

tw_enable_cache(SQLite = FALSE)
tw_set_cache_db(driver = "MySQL",
                host = "localhost",
                port = 3306,
                database = "tidywikidatar",
                user = "secret_username",
                pwd = "secret_password")
```

Get an item

```
tw_search("Mechelen")
```

Show 5 entries Search:

	id	label	description
1	Q162022	Mechelen	city in the province of Antwerp, Belgium
2	Q100020	Mechelen	town in Gulpen-Wittem, Netherlands
3	Q21765751	Mechelen	town in Mechelen municipality, Belgium
4	Q406287	Mechelen	Wikimedia disambiguation page
5	Q3597347	13293 Mechelen	asteroid

Showing 1 to 5 of 10 entries

Previous 1 2 Next

Get an item

```r
tw_search("Mechelen") %>%
  slice(1) %>%
  tw_get()
```

Show 5 entries   Search:

	id	property	value	rank
1	Q162022	alias_en	Mechlin	
2	Q162022	alias_en	Malines	
3	Q162022	description_en	city in the province of Antwerp, Belgium	
4	Q162022	label_en	Mechelen	
5	Q162022	P1036	2--49322	deprecated

```r
tw_search("Mechelen") %>%
  slice(1) %>%
  tw_get() %>%
  tw_label()
```

Show 5 entries Search:

	id	property	value	rank
1	Mechelen		Mechlin	
2	Mechelen		Malines	
3	Mechelen		city in the province of Antwerp, Belgium	
4	Mechelen		Mechelen	
5	Mechelen	Dewey Decimal Classification	2--49322	deprecated

Showing 1 to 5 of 110 entries

Get a specific property

```r
tw_search("European Union") %>%
  slice(1) %>%
  tw_get_property(p = "P31") # instance of
```

Show 5 entries   Search:

	id	property	value	rank
1	Q458	P31	Q4120211	normal
2	Q458	P31	Q3623811	normal
3	Q458	P31	Q1335818	normal
4	Q458	P31	Q7210356	normal
5	Q458	P31	Q1048835	normal

Get a specific property

```r
tw_search("European Union") %>%
  slice(1) %>%
  tw_get_property(p = "P150") %>%  # contains administrative territorial entity
  tw_label()
```

Show 5 entries

Search:

	id	property	value	rank
1	European Union	contains administrative territorial entity	Belgium	preferred
2	European Union	contains administrative territorial entity	Germany	preferred
3	European Union	contains administrative territorial entity	France	preferred

What about qualifiers?

e.g. when did member states join the EU?

https://www.wikidata.org/wiki/Q458#P150

```r
tw_get_qualifiers(id = "Q458", # European Union
                  p = "P150") # contains administrative territorial entity
```

```
## # A tibble: 37 × 8
##    id    property qualifier_id qualifier_prope… qualifier_value qualifier_value…
##    <chr> <chr>    <chr>        <chr>            <chr>           <chr>
##  1 Q458  P150     Q31          P580             +1958-01-01T00… time
##  2 Q458  P150     Q183         P580             +1958-01-01T00… time
##  3 Q458  P150     Q142         P580             +1958-01-01T00… time
##  4 Q458  P150     Q142         P1012            Q3769           wikibase-entity…
##  5 Q458  P150     Q142         P1012            Q17012          wikibase-entity…
##  6 Q458  P150     Q142         P1012            Q17054          wikibase-entity…
##  7 Q458  P150     Q142         P1012            Q17063          wikibase-entity…
##  8 Q458  P150     Q142         P1012            Q17070          wikibase-entity…
##  9 Q458  P150     Q142         P1012            Q126125         wikibase-entity…
## 10 Q458  P150     Q38          P580             +1958-01-01T00… time
## # … with 27 more rows, and 2 more variables: rank <chr>, set <dbl>
```

What about qualifiers?

```
tw_get_qualifiers(id = "Q458", # European Union
                  p = "P150") %>% # contains administrative territorial entity
  filter(qualifier_property == "P580") %>% # keep only "start time"
  transmute(country = tw_get_label(qualifier_id),
            start_time = qualifier_value)
```

```
## # A tibble: 28 × 2
##    country                    start_time
##    <chr>                      <chr>
##  1 Belgium                    +1958-01-01T00:00:00Z
##  2 Germany                    +1958-01-01T00:00:00Z
##  3 France                     +1958-01-01T00:00:00Z
##  4 Italy                      +1958-01-01T00:00:00Z
##  5 Luxembourg                 +1958-01-01T00:00:00Z
##  6 Kingdom of the Netherlands +1958-01-01T00:00:00Z
##  7 Denmark                    +1973-01-01T00:00:00Z
##  8 Republic of Ireland        +1973-01-01T00:00:00Z
##  9 Greece                     +1981-01-01T00:00:00Z
## 10 Portugal                   +1986-01-01T00:00:00Z
## # … with 18 more rows
```

- for more, check `tw_get_property_with_details()`

Dealing with multiple results when only one is needed

Easy questions can be difficult: in which country is London?

```r
tibble::tibble(city_qid = c("Q84")) %>%
  dplyr::mutate(city_label = tw_get_label(city_qid),
                country_qid = tw_get_p(id = city_qid,
                                       p = "P17")) %>%
  tidyr::unnest(country_qid) %>%
  mutate(country = tw_get_label(country_qid))
```

```
## # A tibble: 8 × 4
##   city_qid city_label country_qid country
##   <chr>    <chr>      <chr>       <chr>
## 1 Q84      London     Q2277       Roman Empire
## 2 Q84      London     Q110888     Kingdom of Essex
## 3 Q84      London     Q105092     Kingdom of Mercia
## 4 Q84      London     Q105313     Kingdom of Wessex
## 5 Q84      London     Q179876     Kingdom of England
## 6 Q84      London     Q161885     Great Britain
## 7 Q84      London     Q174193     United Kingdom of Great Britain and Ireland
## 8 Q84      London     Q145        United Kingdom
```

Dealing with multiple results when only one is needed

- keeping first result is tricky
- keeping only preferred may still give more than one result
- people who love tabular data often want just one result, that needs to be "good enough"

```
tibble::tibble(city_qid = c("Q84", "Q220")) %>%
  dplyr::mutate(city_label = tw_get_label(city_qid),
                country_qid = tw_get_p(id = city_qid,
                                       p = "P17",
                                       preferred = TRUE,
                                      # latest_start_time = TRUE,
                                       only_first = TRUE)) %>%
  dplyr::mutate(country_label = tw_get_label(country_qid))
```

```
## # A tibble: 2 × 4
##   city_qid city_label country_qid country_label
##   <chr>    <chr>      <chr>       <chr>
## 1 Q84      London     Q145        United Kingdom
## 2 Q220     Rome       Q38         Italy
```

Less typing

```r
tw_qid_meps %>%
  head() %>%
  mutate(name = tw_get_label(id),
         pob = tw_get_p1(id = id, p = "P19")) %>%
  mutate(pob_label = tw_get_label(id = pob),
         pob_coordinates = tw_get_p1(id = pob, p = "P625"))
```

```
## # A tibble: 6 × 5
##   id    name                 pob     pob_label                  pob_coordinates
##   <chr> <chr>                <chr>   <chr>                      <chr>
## 1 Q157  François Hollande    Q30974  Rouen                      49.44305555555…
## 2 Q329  Nicolas Sarkozy      Q90     Paris                      48.85694444444…
## 3 Q1220 Giorgio Napolitano   Q2634   Naples                     40.83333333333…
## 4 Q1275 Gladwyn Jebb         Q163    Yorkshire                  53.95833333333…
## 5 Q2105 Jacques Chirac       Q238723 5th arrondissement of Paris 48.84722222222…
## 6 Q2512 Kurt Georg Kiesinger Q7019   Albstadt                   48.21194444444…
```

More properties, all at once

Different entry points

Search

- `tw_search()` - search strings

Query

- `tw_query()` - simple queries based on property/value couples
- `tw_get_all_with_p()` - get all items that have a given property, irrespective of their value

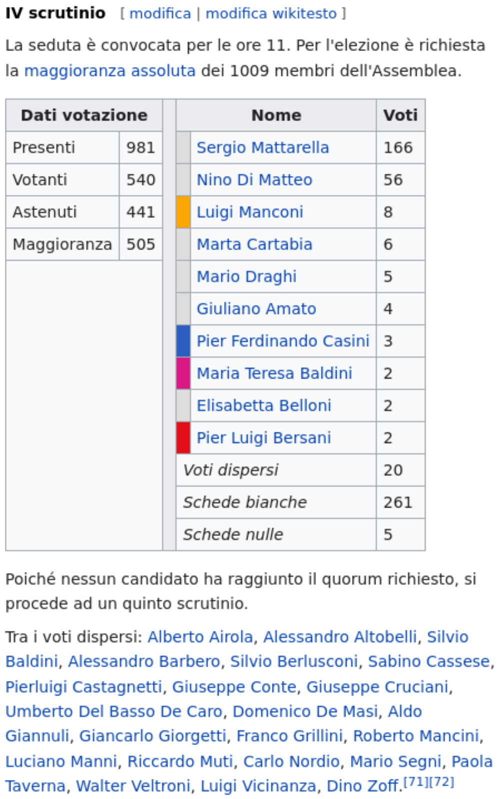
Based on Wikipedia

- `tw_get_wikipedia_page_links()` - Get all Wikidata Q identifiers of all Wikipedia pages linked to input

An example starting from Wikipedia

Election of the President of the Republic in Italy

- Election of the President of the Republic in Italy
- the electoral college can vote literally for whoever they like
- the list ends up including very different candidates, from respected intellectuals to football players and porn actors
- almost all of them with one thing in common: they are on Wikidata, but Wikidata does not know they have something in common.



IV scrutinio [modifica | modifica wikitesto]

La seduta è convocata per le ore 11. Per l'elezione è richiesta la maggioranza assoluta dei 1009 membri dell'Assemblea.

Dati votazione			Nome	Voti
Presenti	981		Sergio Mattarella	166
Votanti	540		Nino Di Matteo	56
Astenuti	441	🟧	Luigi Manconi	8
Maggioranza	505		Marta Cartabia	6
			Mario Draghi	5
			Giuliano Amato	4
		🟦	Pier Ferdinando Casini	3
		🟪	Maria Teresa Baldini	2
			Elisabetta Belloni	2
		🟥	Pier Luigi Bersani	2
			Voti dispersi	20
			Schede bianche	261
			Schede nulle	5

Poiché nessun candidato ha raggiunto il quorum richiesto, si procede ad un quinto scrutinio.

Tra i voti dispersi: Alberto Airola, Alessandro Altobelli, Silvio Baldini, Alessandro Barbero, Silvio Berlusconi, Sabino Cassese, Pierluigi Castagnetti, Giuseppe Conte, Giuseppe Cruciani, Umberto Del Basso De Caro, Domenico De Masi, Aldo Giannuli, Giancarlo Giorgetti, Franco Grillini, Roberto Mancini, Luciano Manni, Riccardo Muti, Carlo Nordio, Mario Segni, Paola Taverna, Walter Veltroni, Luigi Vicinanza, Dino Zoff.[71][72]

Wikidata identifiers

Take a single section:

```
df <- tw_get_wikipedia_page_section_links(
  title = "Elezione del Presidente della Repubblica Ital
  section_title = "IV scrutinio",
  language = "it")

df %>% select(wikipedia_title, qid)
```

wikipedia_title	qid
Adnkronos	Q359875
Alberto Airola	Q14636378
Aldo Giannuli	Q3609233
Alessandro Altobelli	Q346945
Alessandro Barbero	Q960451
Carlo Nordio	Q19357364
Dino Zoff	Q180661
Domenico De Masi	Q3713005

Find out more

```r
pob_df <- df %>%
  select(qid) %>%
  mutate(name = tw_get_label(qid)) %>%
  mutate(place_of_birth_id = tw_get_p(id = qid, p = "P19",only_first = TRUE)) %>%
  mutate(place_of_birth = tw_get_label(place_of_birth_id)) %>%
  mutate(place_of_birth_coordinates = tw_get_p(id = place_of_birth_id,
                                               p = "P625",
                                               only_first = TRUE))
pob_df
```

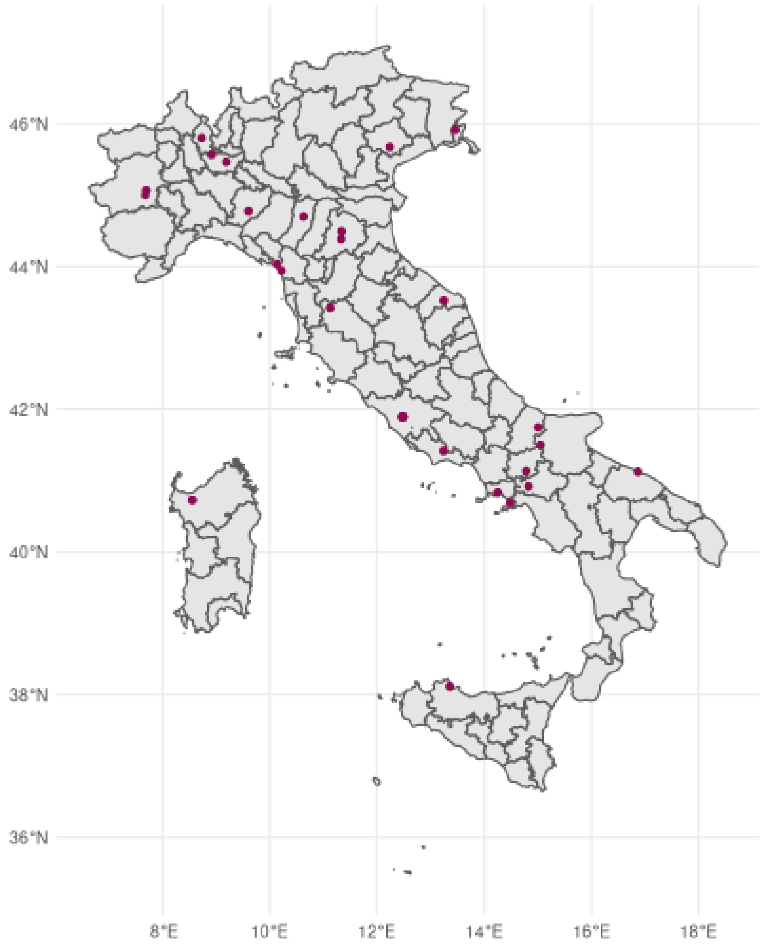
```
## # A tibble: 36 × 5
##    qid       name               place_of_birth_… place_of_birth place_of_birth_…
##    <chr>     <chr>              <chr>            <chr>          <chr>
##  1 Q359875   Adnkronos          <NA>             <NA>           <NA>
##  2 Q14636378 Alberto Airola     Q9474            Moncalieri     45,7.683333
##  3 Q3609233  Aldo Giannuli      Q3519            Bari           41.125277777778…
##  4 Q346945   Alessandro Altobe… Q128211          Sonnino        41.414458333333…
##  5 Q960451   Alessandro Barbero Q495             Turin          45.066666666667…
##  6 Q19357364 Carlo Nordio       Q5475            Treviso        45.672219444444…
##  7 Q180661   Dino Zoff          Q53131           Mariano del F… 45.916666666667…
##  8 Q3713005  Domenico de Masi   Q277969          Rotello        41.7475,15.0041…
##  9 Q3723207  Elisabetta Belloni Q220             Rome           41.893055555556…
```

Here they are on a map

```
pob_sf <- pob_df  %>%
  tidyr::separate(
    col = place_of_birth_coordinates,
    into = c("pob_latitude","pob_longitude"),
    sep = ",",
    remove = TRUE,
    convert = TRUE) %>%
  filter(is.na(pob_latitude)==FALSE) %>%
  sf::st_as_sf(coords = c("pob_longitude", "pob_latitude

library("ggplot2")
pop_gg <- ggplot() +
  geom_sf(data = ll_get_nuts_it(level = 3, no_check_cert
  geom_sf(data = pob_sf, colour = "deeppink4") +
  theme_minimal()
```

```
## ℹ Source: https://www.istat.it/it/archivio/222527
## ℹ Istat (CC-BY)
```



All the usual things we expect from Wikidata

```
occupation_df <- df %>%
  pull(qid) %>%
  tw_get_property(p = "P31") %>% # get "instance of"
  filter(value == "Q5") %>% # keep only humans
  tw_get_property(p = "P106") %>% # get occupation
 # filter(value!="Q82955") %>% # exclude politicians
  group_by(value) %>%
  count(sort = TRUE) %>%
  ungroup() %>%
  transmute(occupation = tw_get_label(value), n)
```

occupation	n
politician	71
university teacher	19
judge	16
lawyer	14
jurist	10
deputy chairperson	9
basketball player	9
physician	9
clerk	9
journalist	7

And other things useful for data visualisation and interactive interfaces, e.g. quick access to images...

```r
president_df <- tw_search("Sergio Mattarella") %>%
  tw_filter_first(p = "P31", q = "Q5")

president_df %>% tw_get_image()
```

```
## # A tibble: 3 × 2
##   id       image
##   <chr>    <chr>
## 1 Q3956186 Sergio Mattarella Presidente della Repubblica Italiana.jpg
## 2 Q3956186 Sergio Mattarella Presidente della Repubblica Italiana.jpg
## 3 Q3956186 Sergio Mattarella Presidente della Repubblica Italiana.jpg
```

```r
president_df %>%  tw_get_image(format = "embed", width = 300) %>% pull(image)
```

```
## [1] "https://commons.wikimedia.org/w/index.php?title=Special:Redirect/file/Sergio Mattarella Presidente della Repubblica Italiana.
## [2] "https://commons.wikimedia.org/w/index.php?title=Special:Redirect/file/Sergio Mattarella Presidente della Repubblica Italiana.
## [3] "https://commons.wikimedia.org/w/index.php?title=Special:Redirect/file/Sergio Mattarella Presidente della Repubblica Italiana.
```

…with metadata and credits line

```
president_df %>% tw_get_image_metadata() %>%
  tidyr::pivot_longer(cols = -1, names_to = "property", values_transform = as.character)
```

```
## # A tibble: 18 × 3
##    id       property                   value
##    <chr>    <chr>                      <chr>
##  1 Q3956186 image_filename             "Sergio Mattarella Presidente della Repu…
##  2 Q3956186 object_name                "Sergio Mattarella Presidente della Repu…
##  3 Q3956186 image_description          "Official picture of the <a href=\"https…
##  4 Q3956186 categories                 "Attribution only license|Images from th…
##  5 Q3956186 assessments                ""
##  6 Q3956186 credit                     "<a rel=\"nofollow\" class=\"external te…
##  7 Q3956186 artist                     "Unknown author<span style=\"display: no…
##  8 Q3956186 permission                  <NA>
##  9 Q3956186 license_short_name         "Attribution"
## 10 Q3956186 license_url                 <NA>
## 11 Q3956186 license                     <NA>
## 12 Q3956186 usage_terms                 <NA>
## 13 Q3956186 attribution_required        <NA>
## 14 Q3956186 copyrighted                 <NA>
## 15 Q3956186 restrictions               "personality"
## 16 Q3956186 date_time                  "2022-05-14 10:47:24"
## 17 Q3956186 date_time_original         "2022-01-29"
## 18 Q3956186 commons_metadata_extension "1.2"
```

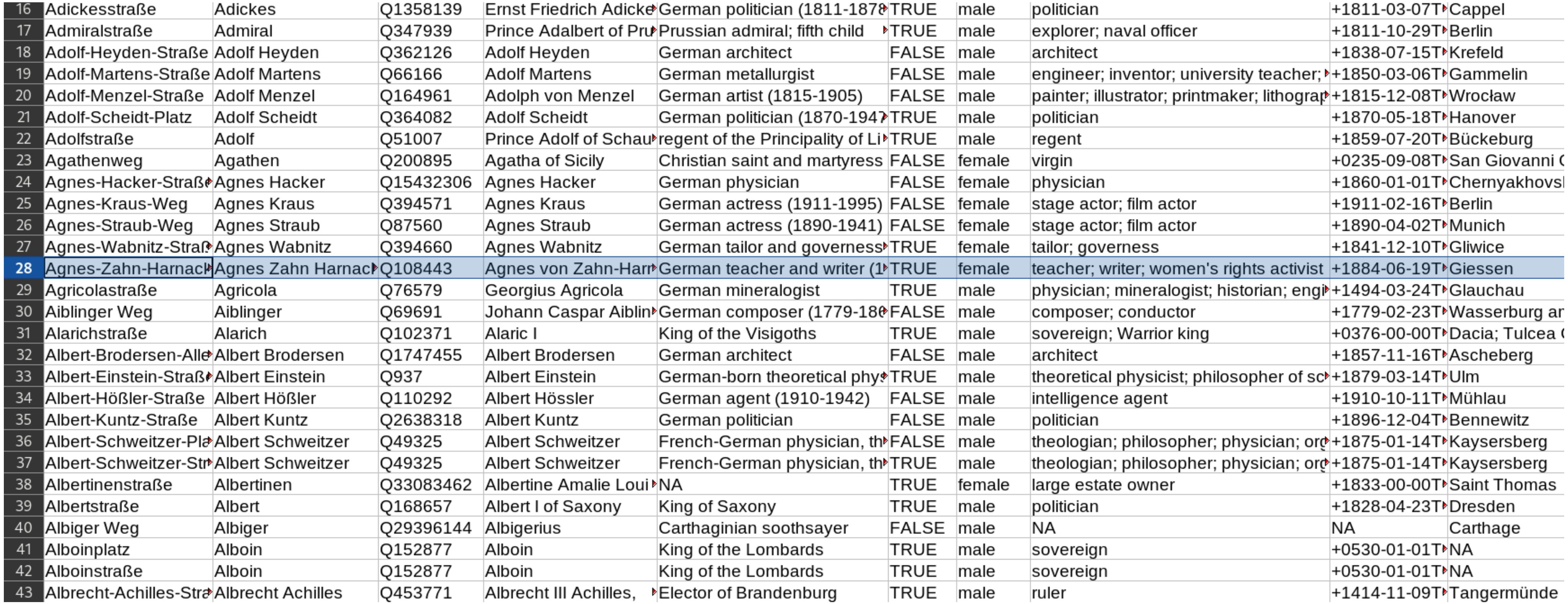
Back and forth between Wikidata and Wikipedia

This gets the Q identifier of all pages linked from a the Wikipedia page of a given Q identifier. Easy peasy :-)

```r
president_df %>%
  tw_get_wikipedia(language = "it") %>% # gets url of Wikipedia page from QID
  tw_get_wikipedia_page_links(language = "it") %>%
  select(wikipedia_title, qid)
```

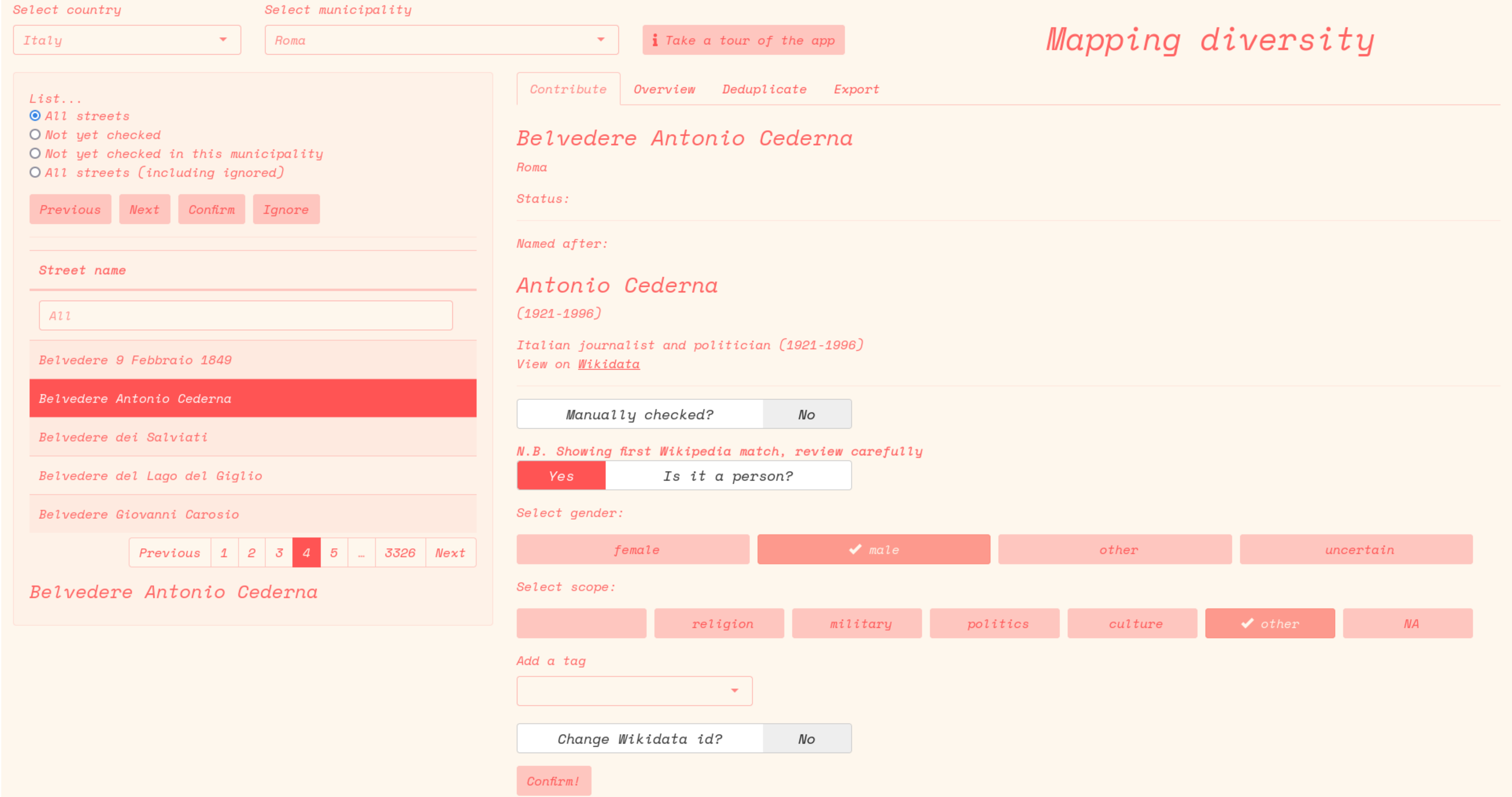
```
## # A tibble: 491 × 2
##    wikipedia_title                             qid
##    <chr>                                       <chr>
##  1 Fabio Vander                                <NA>
##  2 Ordine per Meriti Eccezionali               <NA>
##  3 Discussioni template:Capi di Stato d'Europa <NA>
##  4 1941                                        Q5231
##  5 1987                                        Q2429
##  6 1989                                        Q2425
##  7 1990                                        Q2064
##  8 1998                                        Q2089
##  9 1999                                        Q2091
## 10 2001                                        Q1988
## # … with 481 more rows
```

Starting from strings



16	Adickesstraße	Adickes	Q1358139	Ernst Friedrich Adicke	German politician (1811-1878	TRUE	male	politician	+1811-03-07T	Cappel
17	Admiralstraße	Admiral	Q347939	Prince Adalbert of Pru	Prussian admiral; fifth child	TRUE	male	explorer; naval officer	+1811-10-29T	Berlin
18	Adolf-Heyden-Straße	Adolf Heyden	Q362126	Adolf Heyden	German architect	FALSE	male	architect	+1838-07-15T	Krefeld
19	Adolf-Martens-Straße	Adolf Martens	Q66166	Adolf Martens	German metallurgist	FALSE	male	engineer; inventor; university teacher;	+1850-03-06T	Gammelin
20	Adolf-Menzel-Straße	Adolf Menzel	Q164961	Adolph von Menzel	German artist (1815-1905)	FALSE	male	painter; illustrator; printmaker; lithograp	+1815-12-08T	Wrocław
21	Adolf-Scheidt-Platz	Adolf Scheidt	Q364082	Adolf Scheidt	German politician (1870-1947	TRUE	male	politician	+1870-05-18T	Hanover
22	Adolfstraße	Adolf	Q51007	Prince Adolf of Schau	regent of the Principality of Li	TRUE	male	regent	+1859-07-20T	Bückeburg
23	Agathenweg	Agathen	Q200895	Agatha of Sicily	Christian saint and martyress	FALSE	female	virgin	+0235-09-08T	San Giovanni (
24	Agnes-Hacker-Straße	Agnes Hacker	Q15432306	Agnes Hacker	German physician	FALSE	female	physician	+1860-01-01T	Chernyakhovsk
25	Agnes-Kraus-Weg	Agnes Kraus	Q394571	Agnes Kraus	German actress (1911-1995)	FALSE	female	stage actor; film actor	+1911-02-16T	Berlin
26	Agnes-Straub-Weg	Agnes Straub	Q87560	Agnes Straub	German actress (1890-1941)	FALSE	female	stage actor; film actor	+1890-04-02T	Munich
27	Agnes-Wabnitz-Straß	Agnes Wabnitz	Q394660	Agnes Wabnitz	German tailor and governess	TRUE	female	tailor; governess	+1841-12-10T	Gliwice
28	Agnes-Zahn-Harnack	Agnes Zahn Harnack	Q108443	Agnes von Zahn-Harn	German teacher and writer (1	TRUE	female	teacher; writer; women's rights activist	+1884-06-19T	Giessen
29	Agricolastraße	Agricola	Q76579	Georgius Agricola	German mineralogist	TRUE	male	physician; mineralogist; historian; engi	+1494-03-24T	Glauchau
30	Aiblinger Weg	Aiblinger	Q69691	Johann Caspar Aiblin	German composer (1779-186	FALSE	male	composer; conductor	+1779-02-23T	Wasserburg ar
31	Alarichstraße	Alarich	Q102371	Alaric I	King of the Visigoths	TRUE	male	sovereign; Warrior king	+0376-00-00T	Dacia; Tulcea (
32	Albert-Brodersen-Alle	Albert Brodersen	Q1747455	Albert Brodersen	German architect	FALSE	male	architect	+1857-11-16T	Ascheberg
33	Albert-Einstein-Straße	Albert Einstein	Q937	Albert Einstein	German-born theoretical phys	TRUE	male	theoretical physicist; philosopher of sc	+1879-03-14T	Ulm
34	Albert-Hößler-Straße	Albert Hößler	Q110292	Albert Hössler	German agent (1910-1942)	FALSE	male	intelligence agent	+1910-10-11T	Mühlau
35	Albert-Kuntz-Straße	Albert Kuntz	Q2638318	Albert Kuntz	German politician	FALSE	male	politician	+1896-12-04T	Bennewitz
36	Albert-Schweitzer-Pla	Albert Schweitzer	Q49325	Albert Schweitzer	French-German physician, th	FALSE	male	theologian; philosopher; physician; org	+1875-01-14T	Kaysersberg
37	Albert-Schweitzer-Str	Albert Schweitzer	Q49325	Albert Schweitzer	French-German physician, th	TRUE	male	theologian; philosopher; physician; org	+1875-01-14T	Kaysersberg
38	Albertinenstraße	Albertinen	Q33083462	Albertine Amalie Loui	NA	TRUE	female	large estate owner	+1833-00-00T	Saint Thomas
39	Albertstraße	Albert	Q168657	Albert I of Saxony	King of Saxony	TRUE	male	politician	+1828-04-23T	Dresden
40	Albiger Weg	Albiger	Q29396144	Albigerius	Carthaginian soothsayer	FALSE	male	NA	NA	Carthage
41	Alboinplatz	Alboin	Q152877	Alboin	King of the Lombards	TRUE	male	sovereign	+0530-01-01T	NA
42	Alboinstraße	Alboin	Q152877	Alboin	King of the Lombards	TRUE	male	sovereign	+0530-01-01T	NA
43	Albrecht-Achilles-Stra	Albrecht Achilles	Q453771	Albrecht III Achilles,	Elector of Brandenburg	TRUE	male	ruler	+1414-11-09T	Tangermünde

Integration with interactive interfaces



Select country

Italy

Select municipality

Roma

Take a tour of the app

Mapping diversity

List...

● All streets
○ Not yet checked
○ Not yet checked in this municipality
○ All streets (including ignored)

Previous | Next | Confirm | Ignore

Street name

All

Belvedere 9 Febbraio 1849

Belvedere Antonio Cederna

Belvedere dei Salviati

Belvedere del Lago del Giglio

Belvedere Giovanni Carosio

Previous | 1 | 2 | 3 | 4 | 5 | ... | 3326 | Next

Belvedere Antonio Cederna

Contribute | Overview | Deduplicate | Export

Belvedere Antonio Cederna

Roma

Status:

Named after:

Antonio Cederna

(1921-1996)

Italian journalist and politician (1921-1996)

View on Wikidata

Manually checked? No

N.B. Showing first Wikipedia match, review carefully

Yes Is it a person?

Select gender:

female | ✓ male | other | uncertain

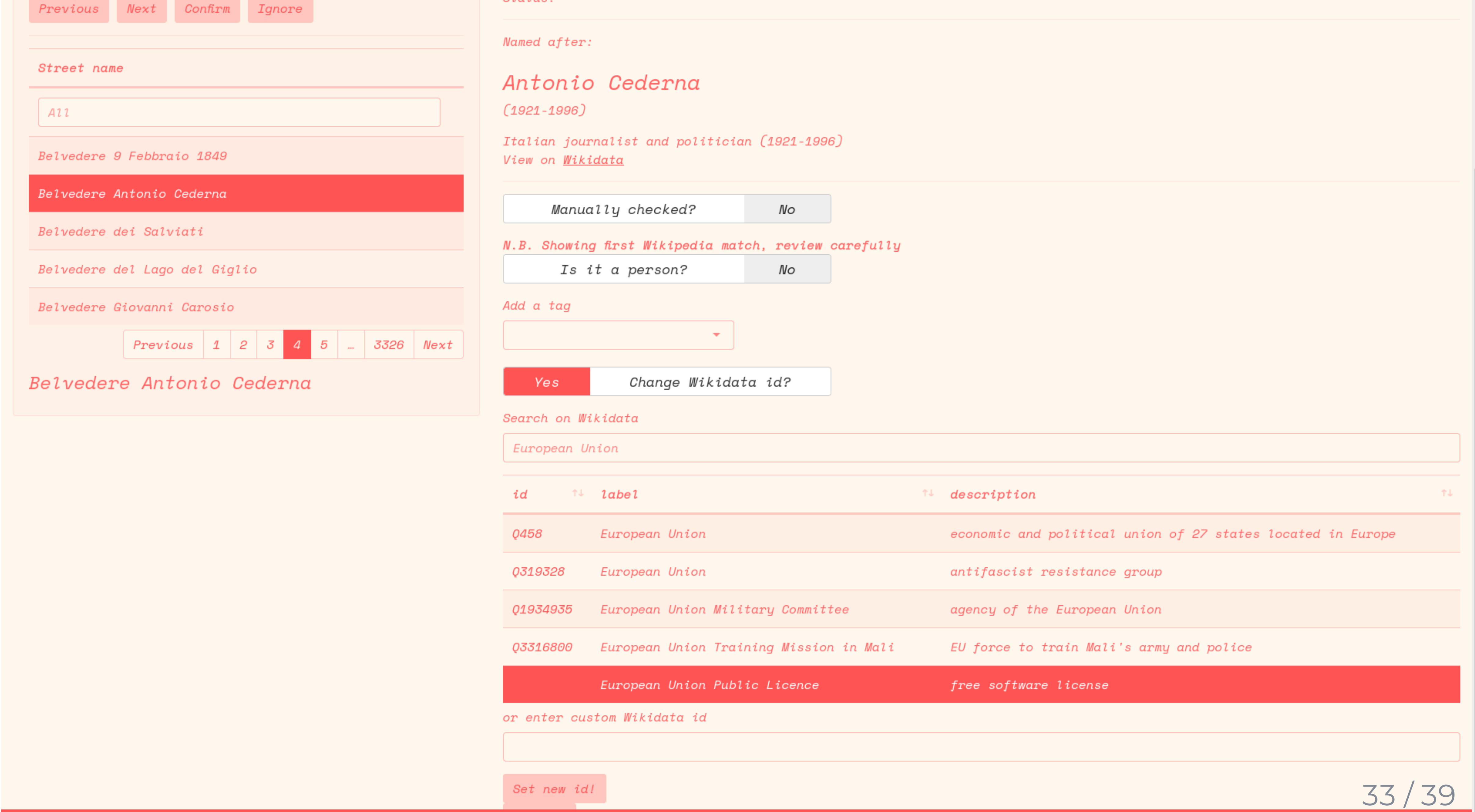
Select scope:

| religion | military | politics | culture | ✓ other | NA

Add a tag

Change Wikidata id? No

Confirm!



Street name

All

- Belvedere 9 Febbraio 1849
- Belvedere Antonio Cederna
- Belvedere dei Salviati
- Belvedere del Lago del Giglio
- Belvedere Giovanni Carosio

Belvedere Antonio Cederna

Named after:

Antonio Cederna

(1921-1996)

Italian journalist and politician (1921-1996)
View on Wikidata

Manually checked? No

N.B. Showing first Wikipedia match, review carefully

Is it a person? No

Add a tag

Yes Change Wikidata id?

Search on Wikidata

European Union

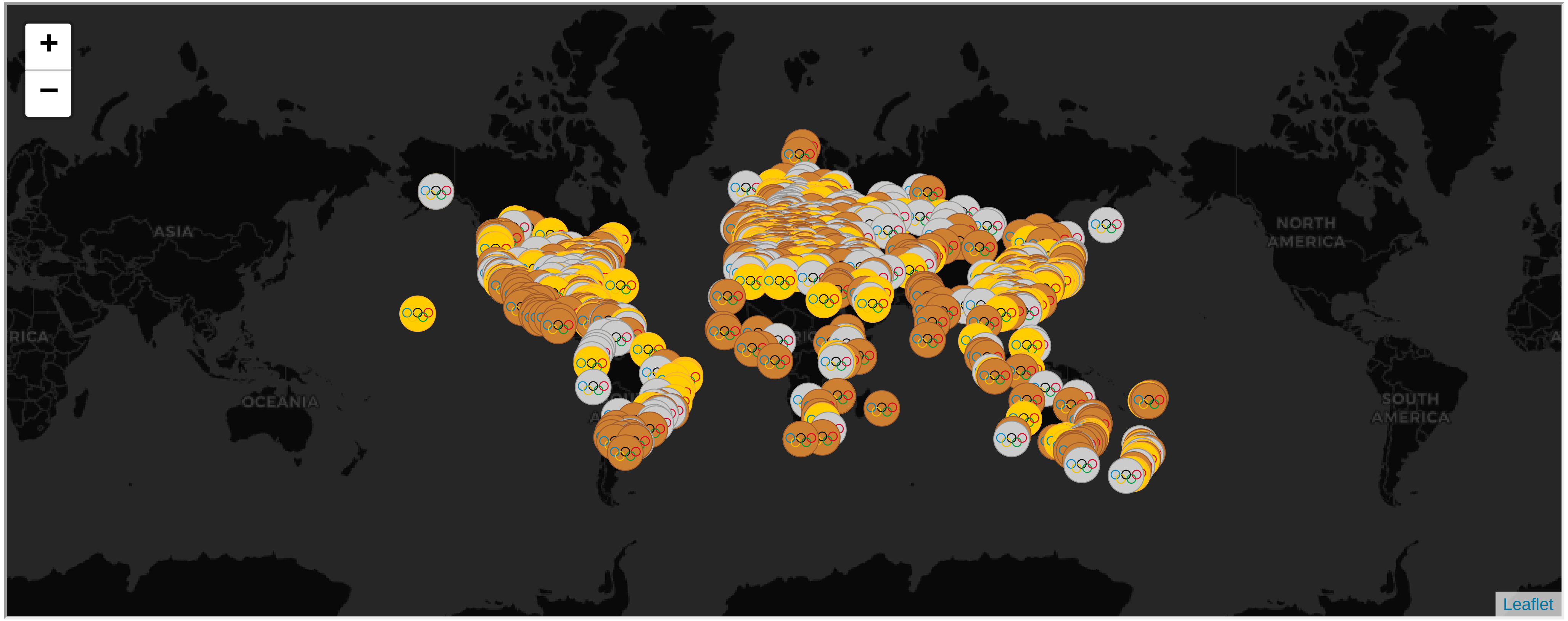
id	label	description
Q458	European Union	economic and political union of 27 states located in Europe
Q319328	European Union	antifascist resistance group
Q1934935	European Union Military Committee	agency of the European Union
Q3316800	European Union Training Mission in Mali	EU force to train Mali's army and police
	European Union Public Licence	free software license

or enter custom Wikidata id

A couple of examples of practical use cases

Olympics 2020 medalists by place of birth

https://github.com/EDJNet/olympics2020nuts

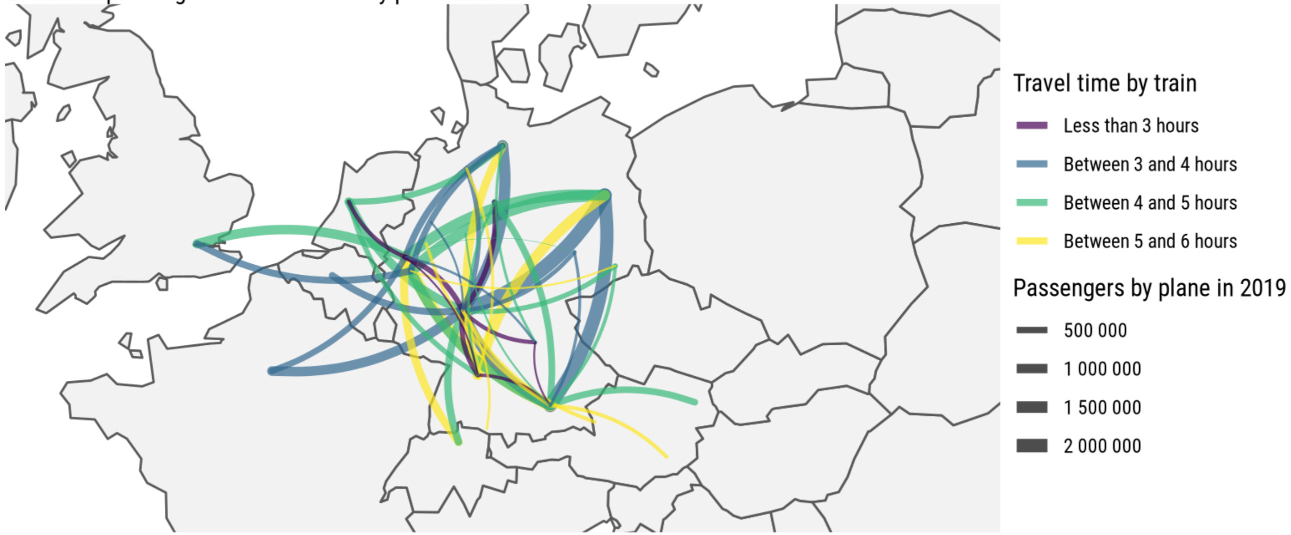


Main air routes that could be travelled by train

Wikidata used for: defining city hubs for airports, getting coordinates of airports (for excluding those on islands), use unique identifiers for merging with train dataset https://edjnet.github.io/european_routes/

All these routes involving Germany take less than 6 hours by train

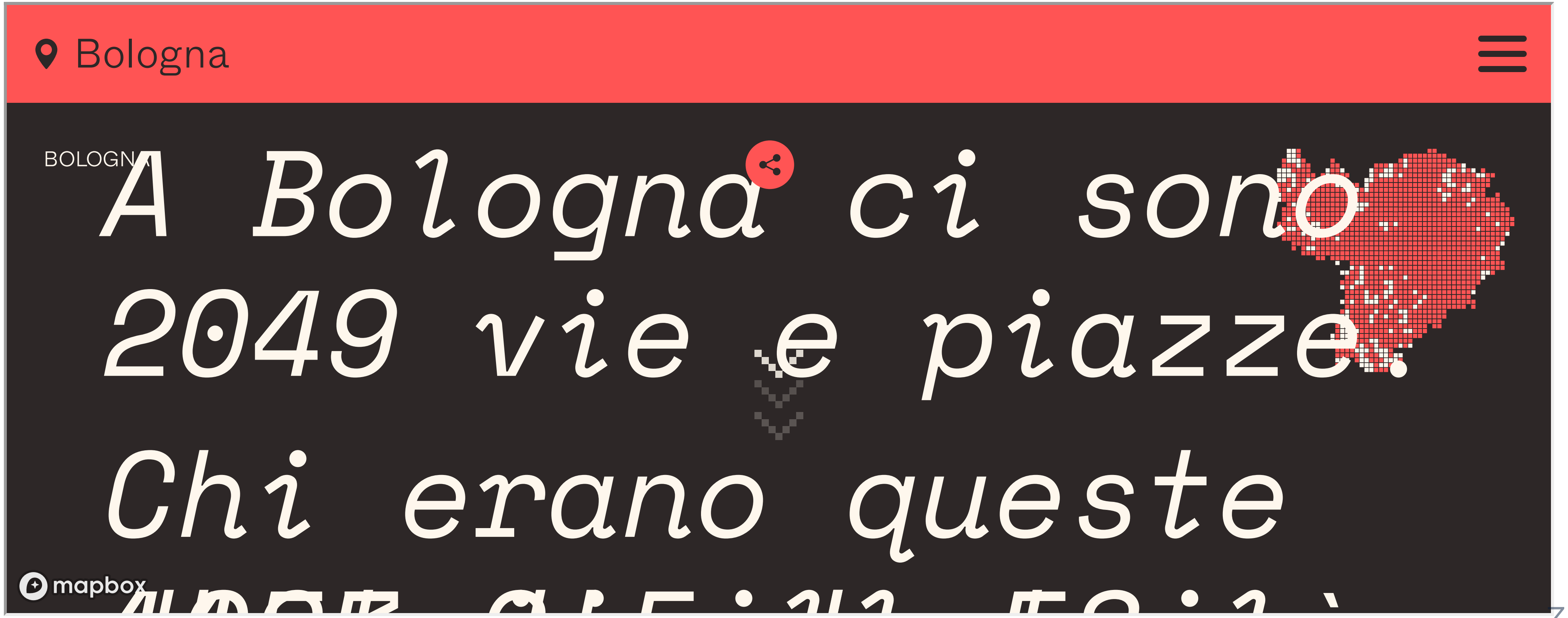
28 million passengers travelled them by plane in 2019



Giorgio Comai / @EdjNet | Source: Eurostat `avia_par_`, OBCT for Greenpeace

Mapping diversity

https://medium.com/european-data-journalism-network/finding-gendered-street-names-a-step-by-step-walkthrough-with-r-7608c2d36a77



Bologna

BOLOGN

A Bologna ci sono 2049 vie e piazze. Chi erano queste

General issues

- if you are processing many thousands of items, the current approach can be very slow when run for the first time (near instant thanks to caching later)
    - no obvious long term solution, but a future version will allow for an easier way to share the cache to make sure others can also run the script instantly
- no easy way to "give back" to Wikidata

`tidywikidatar`

Check out website with documentation and examples:
https://edjnet.github.io/tidywikidatar/

CRAN 0.5.2 | downloads 406/month | downloads 5783

- everything in tabular format
- one row, one piece of information
- easy local caching
- easy integration with `dplyr` piped routines
- get image credits from WikiMedia commons
- include Wikipedia in the exploration, or use it as a starting point