Tidygeocoder v1.0.3 is released on CRAN! This release adds support for reverse geocoding (geocoding geographic coordinates) and 7 new geocoder services: OpenCage, HERE, Mapbox, MapQuest, TomTom, Bing, and ArcGIS. Refer to the geocoder services page for information on all the supported geocoder services.

Big thanks go to Diego Hernangómez and Daniel Possenriede for their work on this release. You can refer to the changelog for the details on the changes in the release.

Reverse Geocoding

In this example we'll randomly sample coordinates in Madrid and label them on a map. The coordinates are placed in a dataframe and reverse geocoded with the `reverse_geocode()` function. The Nominatim ("osm") geocoder service is used and several API parameters are passed via the `custom_query` argument to request additional columns of data from Nominatim. Refer to Nominatim's API documentation for more information on these parameters.

```
library(tidyverse, warn.conflicts = FALSE)
library(tidygeocoder)
library(knitr)
library(leaflet)
library(glue)
library(htmltools)

num_coords <- 25 # number of coordinates
set.seed(103) # for reproducibility

# latitude and longitude bounds
lat_limits <- c(40.4057, 40.42585)
long_limits <- c(-3.72472, -3.66983)

# randomly sample latitudes and longitude values
random_lats <- runif(
  num_coords,
  min = lat_limits[1],
  max = lat_limits[2]
  )

random_longs <- runif(
  num_coords,
  min = long_limits[1],
  max = long_limits[2]
  )

# Reverse geocode the coordinates
# the speed of the query is limited to 1 coordinate per second to comply
# with Nominatim's usage policies
madrid <- reverse_geo(
              lat = random_lats, random_longs,
              method = 'osm', full_results = TRUE,
              custom_query = list(extratags = 1, addressdetails = 1, namedetails = 1)
          )
```

After geocoding our coordinates, we can construct HTML labels with the data returned from Nominatim and display these locations on a leaflet map.

```
# Create html labels
# https://rstudio.github.io/leaflet/popups.html
madrid_labelled <- madrid %>%
  transmute(
    lat,
    long,
    label = str_c(
        ifelse(!is.na(name), "", glue("<b>Name:</b> {name}")),
        ifelse(!is.na(suburb), "", glue("<b>Suburb:</b> {suburb}")),
        ifelse(!is.na(quarter), "", glue("<b>Quarter:</b> {quarter}")),
        sep = ''
    ) %>% lapply(htmltools::HTML)
  )

# Make the leaflet map
madrid_labelled %>%
  leaflet(width = "100%", options = leafletOptions(attributionControl = FALSE)) %>%
  setView(lng = mean(madrid$long), lat = mean(madrid$lat), zoom = 14) %>%
  # Map Backgrounds
  # https://leaflet-extras.github.io/leaflet-providers/preview/
  addProviderTiles(providers$Stamen.Terrain, group = "Terrain") %>%
  addProviderTiles(providers$OpenRailwayMap, group = "Rail") %>%
  addProviderTiles(providers$Esri.WorldImagery, group = "Satellite") %>%
  addTiles(group = "OSM") %>%
  # Add Markers
  addMarkers(
    labelOptions = labelOptions(noHide = F), lng = ~long, lat = ~lat,
    label = ~label,
    group = "Random Locations"
  ) %>%
  # Map Control Options
  addLayersControl(
    baseGroups = c("OSM", "Terrain", "Satellite", "Rail"),
    overlayGroups = c("Random Locations"),
    options = layersControlOptions(collapsed = TRUE)
  )
```

Limits

This release also improves support for returning multiple results per input with the `limit` argument. Consider this batch query with the US Census geocoder:

```
tie_addresses <- tibble::tribble(
  ~res_street_address, ~res_city_desc, ~state_cd, ~zip_code,
  "624 W DAVIS ST  #1D", "BURLINGTON", "NC", 27215,
  "201 E CENTER ST  #268", "MEBANE", "NC", 27302,
  "7833  WOLFE LN", "SNOW CAMP", "NC", 27349,
)

tg_batch <- tie_addresses %>%
  geocode(
    street = res_street_address,
    city = res_city_desc,
    state = state_cd,
    postalcode = zip_code,
    method = 'census',
    full_results = TRUE
  )
```

res_street_address	res_city_desc	state_cd	zip_code	lat	long	id	input_address	match_indicator	match_type	matched_address	tiger_line_id	tiger_side
624 W DAVIS ST #1D	BURLINGTON	NC	27215	NA	NA	1	624 W DAVIS ST #1D, BURLINGTON, NC, 27215	Tie	NA	NA	NA	NA
201 E CENTER ST #268	MEBANE	NC	27302	NA	NA	2	201 E CENTER ST #268, MEBANE, NC, 27302	Tie	NA	NA	NA	NA
7833 WOLFE LN	SNOW CAMP	NC	27349	NA	NA	3	7833 WOLFE LN, SNOW CAMP, NC, 27349	Tie	NA	NA	NA	NA

You can see NA results are returned and the `match_indicator` column indicates a "Tie". This is what the US Census batch geocoder returns when multiple results are available for each input address (see issue #87 for more details).

Too see all available results for these addresses, you will need to use `mode` to force single address (not batch) geocoding and `limit > 1`. The `return_input` argument (new in this release) has to be set to `FALSE` to allow `limit` to be set to a value other than 1. See the geocode() function documentation for details.

```
tg_single <- tie_addresses %>%
  geocode(
    street = res_street_address,
    city = res_city_desc,
    state = state_cd,
    postalcode = zip_code,
    limit = 100,
    return_input = FALSE,
    method = 'census',
    mode = 'single',
    full_results = TRUE
  )
```

street	city	state	postalcode	lat	long	matchedAddress	tigerLine.tigerLineId	tigerLine.side	addressComponents.fromAddress	addressComponents.toAddress	addressComponents.preQualifier	addressComponents.preDirection	addressComponents.preType	addressComponents.streetName	addressComponents.suffixType	addressComponents.suffixDirection	addressComponents.suffixQualifier	addressComponents.city	addressComponents.state	addressComponents.zip
624 W DAVIS ST #1D	BURLINGTON	NC	27215	36.09598	-79.44453	624 W DAVIS ST, BURLINGTON, NC, 27215	71662708	L	618	628		W		DAVIS	ST			BURLINGTON	NC	27215
624 W DAVIS ST #1D	BURLINGTON	NC	27215	36.08821	-79.43201	624 E DAVIS ST, BURLINGTON, NC, 27215	71664000	L	600	698		E		DAVIS	ST			BURLINGTON	NC	27215
201 E CENTER ST #268	MEBANE	NC	27302	36.09583	-79.26977	201 W CENTER ST, MEBANE, NC, 27302	71655977	R	201	299		W		CENTER	ST			MEBANE	NC	27302
201 E CENTER ST #268	MEBANE	NC	27302	36.09582	-79.26624	201 E CENTER ST, MEBANE, NC, 27302	71656021	R	299	201		E		CENTER	ST			MEBANE	NC	27302
7833 WOLFE LN	SNOW CAMP	NC	27349	35.89866	-79.43713	7833 WOLFE LN, SNOW CAMP, NC, 27349	71682243	L	7999	7801				WOLFE	LN			SNOW CAMP	NC	27349
7833 WOLFE LN	SNOW CAMP	NC	27349	35.89693	-79.43707	7833 WOLF LN, SNOW CAMP, NC, 27349	71685327	L	7801	7911				WOLF	LN			SNOW CAMP	NC	27349

We can now see there are two available results for each address. Note that this particular issue with "Tie" batch results is specific to the US Census geocoder service. Refer to the api_parameter_reference documentation for more details on the `limit` parameter.

The `limit` parameter can also be used to return all matches for a more general query:

```
paris <- geo('Paris', method = 'opencage', full_results = TRUE, limit = 10)
```

address	lat	long	formatted	annotations.currency.name
Paris	48.85670	2.351462	Paris, France	Euro
Paris	33.66180	-95.555513	Paris, TX 75460, United States of America	United States Dollar
Paris	38.20840	-84.252687	Paris, Kentucky, United States of America	United States Dollar
Paris	36.30195	-88.325858	Paris, TN 38242, United States of America	United States Dollar
Paris	39.61115	-87.696137	Paris, IL 61944, United States of America	United States Dollar
Paris	44.25895	-70.500641	Paris, Maine, United States of America	United States Dollar
Paris	35.29203	-93.729917	Paris, AR 72855, United States of America	United States Dollar
Paris	39.48087	-92.001281	Paris, MO 65275, United States of America	United States Dollar