MWSUG 2018 - Paper SP-057

# Macro that Can Get Geo Coding Information from the Google Maps API

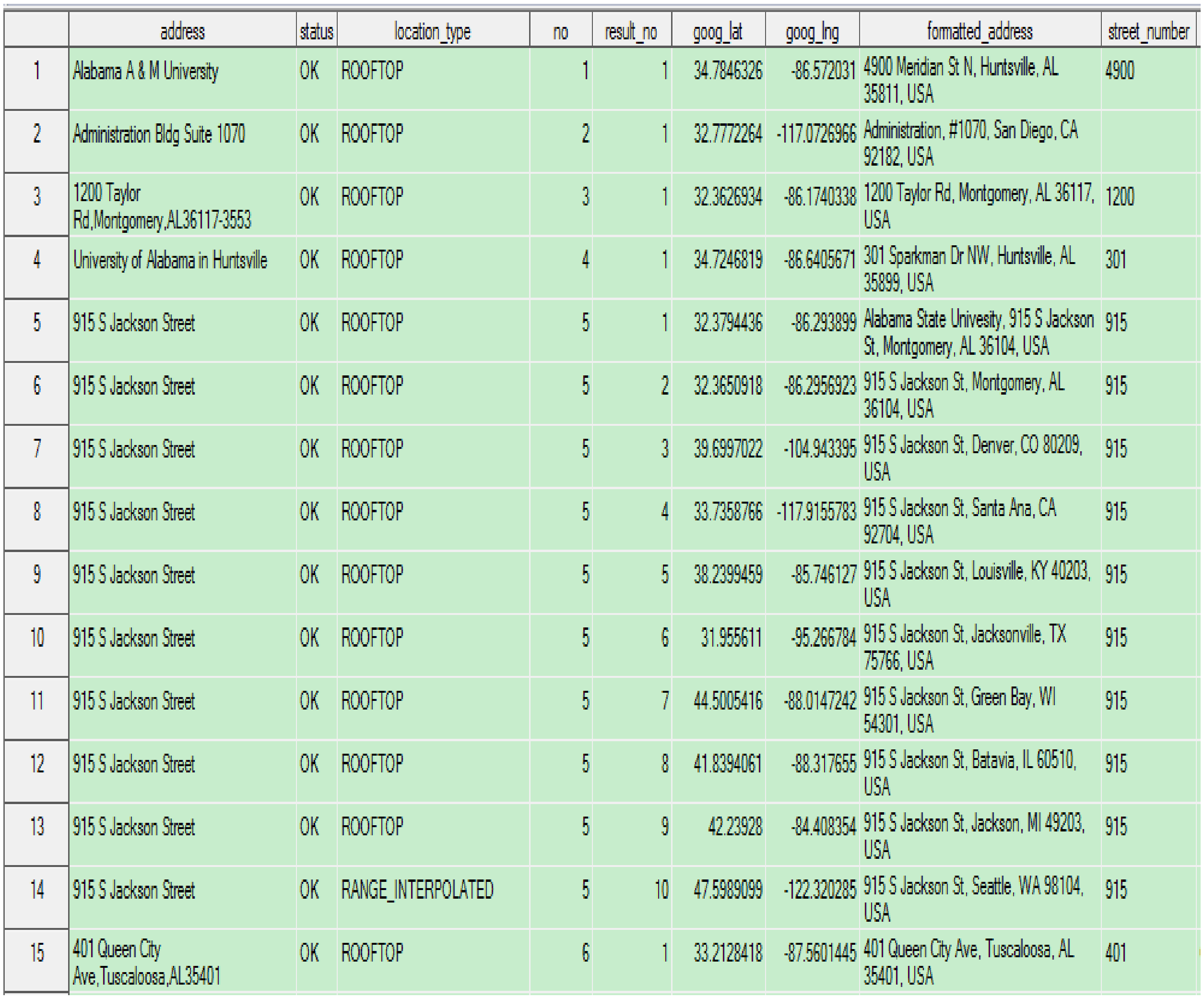
Ting Sa, Cincinnati Children's Hospital Medical Center, Cincinnati, OH

## ABSTRACT

This paper introduces a macro that can automatically get the geo coding information from the Google Maps API for the user. The macro can get the longitude, latitude, standard address, and address components like street number, street name, county or city name, state name, ZIP codes, and so on for the user. To use the macro, the user needs to provide only simple SAS® input data. The macro then automatically gets the data and saves it to a SAS data set for the user. This paper includes all the SAS codes for the macro and provides the input data example to show you how to use the macro.

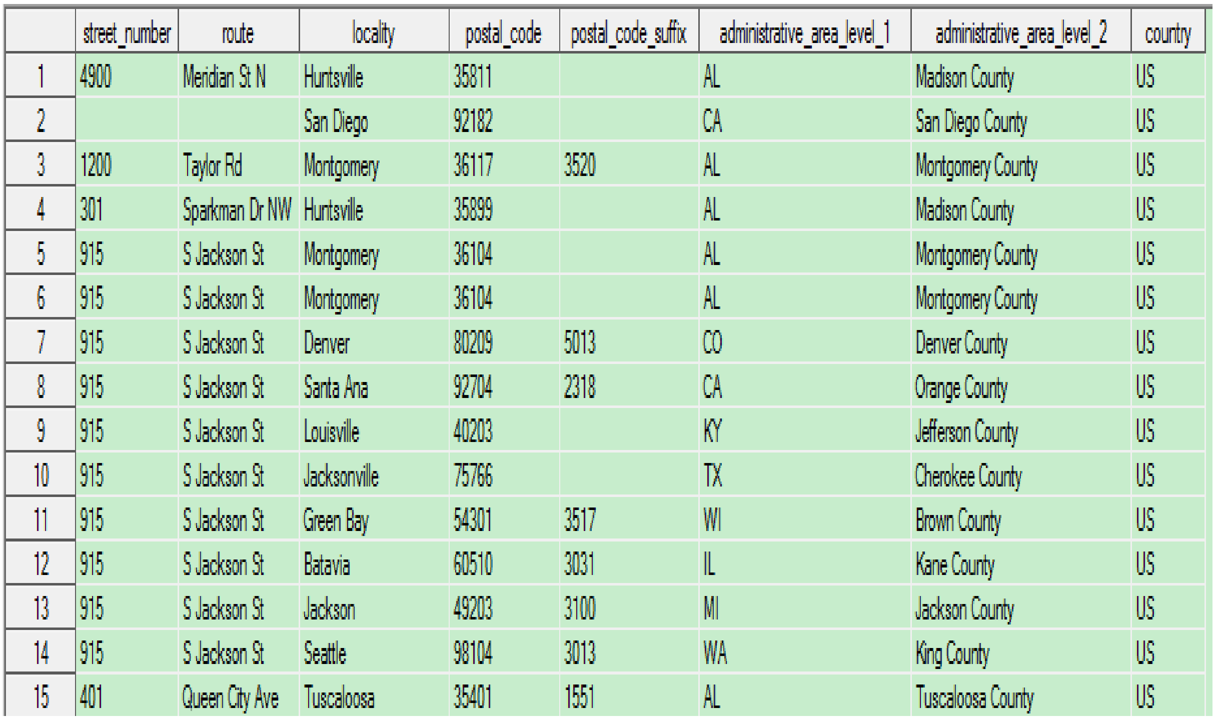
## INTRODUCTION

In this paper a macro is introduced that can help the user to get the geo coding information from the Google map API. Figure 1, Figure 2 are the screenshots for a sample output SAS data set. I will give more detailed explanation for the output data set in the "The Output SAS Data Set" section.



| | address | status | location_type | no | result_no | goog_lat | goog_lng | formatted_address | street_number |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Alabama A & M University | OK | ROOFTOP | 1 | 1 | 34.7846326 | -86.572031 | 4900 Meridian St N, Huntsville, AL 35811, USA | 4900 |
| 2 | Administration Bldg Suite 1070 | OK | ROOFTOP | 2 | 1 | 32.7772264 | -117.0726966 | Administration, #1070, San Diego, CA 92182, USA | |
| 3 | 1200 Taylor Rd,Montgomery,AL36117-3553 | OK | ROOFTOP | 3 | 1 | 32.3626934 | -86.1740338 | 1200 Taylor Rd, Montgomery, AL 36117, USA | 1200 |
| 4 | University of Alabama in Huntsville | OK | ROOFTOP | 4 | 1 | 34.7246819 | -86.6405671 | 301 Sparkman Dr NW, Huntsville, AL 35899, USA | 301 |
| 5 | 915 S Jackson Street | OK | ROOFTOP | 5 | 1 | 32.3794436 | -86.293899 | Alabama State Univesity, 915 S Jackson St, Montgomery, AL 36104, USA | 915 |
| 6 | 915 S Jackson Street | OK | ROOFTOP | 5 | 2 | 32.3650918 | -86.2956923 | 915 S Jackson St, Montgomery, AL 36104, USA | 915 |
| 7 | 915 S Jackson Street | OK | ROOFTOP | 5 | 3 | 39.6997022 | -104.943395 | 915 S Jackson St, Denver, CO 80209, USA | 915 |
| 8 | 915 S Jackson Street | OK | ROOFTOP | 5 | 4 | 33.7358766 | -117.9155783 | 915 S Jackson St, Santa Ana, CA 92704, USA | 915 |
| 9 | 915 S Jackson Street | OK | ROOFTOP | 5 | 5 | 38.2399459 | -85.746127 | 915 S Jackson St, Louisville, KY 40203, USA | 915 |
| 10 | 915 S Jackson Street | OK | ROOFTOP | 5 | 6 | 31.955611 | -95.266784 | 915 S Jackson St, Jacksonville, TX 75766, USA | 915 |
| 11 | 915 S Jackson Street | OK | ROOFTOP | 5 | 7 | 44.5005416 | -88.0147242 | 915 S Jackson St, Green Bay, WI 54301, USA | 915 |
| 12 | 915 S Jackson Street | OK | ROOFTOP | 5 | 8 | 41.8394061 | -88.317655 | 915 S Jackson St, Batavia, IL 60510, USA | 915 |
| 13 | 915 S Jackson Street | OK | ROOFTOP | 5 | 9 | 42.23928 | -84.408354 | 915 S Jackson St, Jackson, MI 49203, USA | 915 |
| 14 | 915 S Jackson Street | OK | RANGE_INTERPOLATED | 5 | 10 | 47.5989099 | -122.320285 | 915 S Jackson St, Seattle, WA 98104, USA | 915 |
| 15 | 401 Queen City Ave,Tuscaloosa,AL35401 | OK | ROOFTOP | 6 | 1 | 33.2128418 | -87.5601445 | 401 Queen City Ave, Tuscaloosa, AL 35401, USA | 401 |

**Figure 1. Part of the Output SAS Data Set Result**



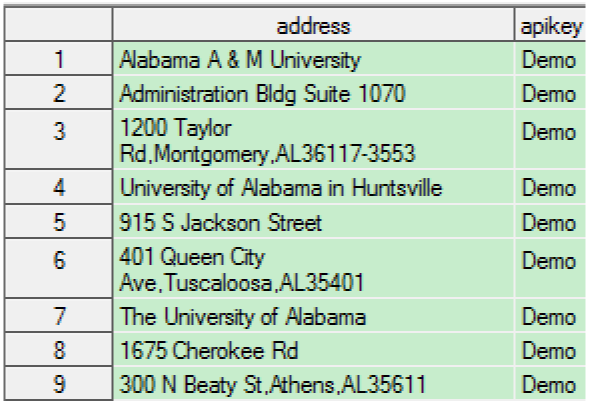
| | street_number | route | locality | postal_code | postal_code_suffix | administrative_area_level_1 | administrative_area_level_2 | country |
|---|---|---|---|---|---|---|---|---|
| 1 | 4900 | Meridian St N | Huntsville | 35811 | | AL | Madison County | US |
| 2 | | | San Diego | 92182 | | CA | San Diego County | US |
| 3 | 1200 | Taylor Rd | Montgomery | 36117 | 3520 | AL | Montgomery County | US |
| 4 | 301 | Sparkman Dr NW | Huntsville | 35899 | | AL | Madison County | US |
| 5 | 915 | S Jackson St | Montgomery | 36104 | | AL | Montgomery County | US |
| 6 | 915 | S Jackson St | Montgomery | 36104 | | AL | Montgomery County | US |
| 7 | 915 | S Jackson St | Denver | 80209 | 5013 | CO | Denver County | US |
| 8 | 915 | S Jackson St | Santa Ana | 92704 | 2318 | CA | Orange County | US |
| 9 | 915 | S Jackson St | Louisville | 40203 | | KY | Jefferson County | US |
| 10 | 915 | S Jackson St | Jacksonville | 75766 | | TX | Cherokee County | US |
| 11 | 915 | S Jackson St | Green Bay | 54301 | 3517 | WI | Brown County | US |
| 12 | 915 | S Jackson St | Batavia | 60510 | 3031 | IL | Kane County | US |
| 13 | 915 | S Jackson St | Jackson | 49203 | 3100 | MI | Jackson County | US |
| 14 | 915 | S Jackson St | Seattle | 98104 | 3013 | WA | King County | US |
| 15 | 401 | Queen City Ave | Tuscaloosa | 35401 | 1551 | AL | Tuscaloosa County | US |

**Figure 2. Part of the Output SAS Data Set Result**

To use the macro, the user needs to provide a simple SAS input data and also needs to get a Google map API key which is free and easy to get from the Google website. To know how to get the API key, check this web address https://developers.google.com/maps/documentation/javascript/get-api-key for more information. In this paper, the macro uses the newest json libname engine to parse the json data sent by the API to get the geo coding information. All the macro SAS codes will be included at the end of the paper.

## THE INPUT DATA SET FOR THE MACRO

Figure 3 shows the structure of the input data set that can be used by the macro. The column "address" contains the address information. The "apikey" column contain the map API key provided by the Google.



| | address | apikey |
|---|---|---|
| 1 | Alabama A & M University | Demo |
| 2 | Administration Bldg Suite 1070 | Demo |
| 3 | 1200 Taylor Rd,Montgomery,AL36117-3553 | Demo |
| 4 | University of Alabama in Huntsville | Demo |
| 5 | 915 S Jackson Street | Demo |
| 6 | 401 Queen City Ave,Tuscaloosa,AL35401 | Demo |
| 7 | The University of Alabama | Demo |
| 8 | 1675 Cherokee Rd | Demo |
| 9 | 300 N Beaty St,Athens,AL35611 | Demo |

**Figure 3. A Sample Input Data Set for the Macro**

The sample input data set is the "Higher Education Datasets" I've downloaded from the website https://inventory.data.gov/dataset/032e19b4-5a90-41dc-83ff-6e4cd234f565/resource/38625c3d-5388-4c16-a30f-d105432553a4. It is a free data set and you can click the link to download the CSV data file “higheducationdata.csv”. Below are the SAS codes I’ve used to create the sample input data set using the downloaded csv file.

```
PROC IMPORT OUT=education
DATAFILE= "C:\higheducationdata.csv"
DBMS=CSV REPLACE;
GETNAMES=YES;
DATAROW=2;
RUN;

data test;
set education;
if _n_ <=30;
address_full=catx(",",ADDR,CITY,cats(STABBR,zip));
length address $100.;
if mod(_n_,3)=1 then address=INSTNM;
else if mod(_n_,3)=2 then address=ADDR;
else if mod(_n_,3)=0 then address=address_full;
keep INSTNM ADDR address_full address LONGITUD LATITUDE;
run;

data testdata;
set test;
apikey="Demo";
keep address apikey;
run;
```

Once you get your own API key, you can replace the “Demo” with your API key value.

## THE GETGEOINFO MACRO

You can find all the macro codes at the end of the paper. Immediately below is the structure of the macro:

```
%macro getGeoInfo(libnm=,datanm=);
```

- The “libnm” is used to indicate the library name for the input dataset.
- The “datanm” is used to indicate the input SAS dataset names.

Below is an example showing you how to call the macro:

```
%getGeoInfo(libnm=work,datanm=testdata);
```

The result of the geo coding information will be saved into a SAS data set called “result_geocode” which can be found in the “work” library.

## THE OUTPUT SAS DATA SET

The output SAS data set contains the following variables in the data set. Below are the descriptions for each variable:

a. The “address” contains the address from the input data set.
b. The “status” contains the status of the request, and may contain the following values:
    - "OK" indicates that no errors occurred; the address was successfully parsed and at least one geocode was returned.
    - "ZERO_RESULTS" indicates that the geocode was successful but returned no results. This may occur if the geocoder was passed a non-existent address.
    - "OVER_QUERY_LIMIT" indicates that you are over your quota.
    - "REQUEST_DENIED" indicates that your request was denied.
    - "INVALID_REQUEST" generally indicates that the query (address, components or latlng) is missing.

    - "UNKNOWN_ERROR" indicates that the request could not be processed due to a server error. The request may succeed if you try again.

c. The “location_type” stores additional data about the specified location and may contain the following values:
    - "ROOFTOP" indicates that the returned result is a precise geocode for which we have location information accurate down to street address precision.
    - "RANGE_INTERPOLATED" indicates that the returned result reflects an approximation (usually on a road) interpolated between two precise points (such as intersections). Interpolated results are generally returned when rooftop geocodes are unavailable for a street address.
    - "GEOMETRIC_CENTER" indicates that the returned result is the geometric center of a result such as a polyline (for example, a street) or polygon (region).
    - "APPROXIMATE" indicates that the returned result is approximate.

d. The “no” contains the address row number in the input data set.<br>
e. The “result_no” contains the parsed address counting number. Because for a given address, the API may return several results, this variable is created to indicate multiple returns.<br>
f. The “goog_lat” contains the latitude information.<br>
g. The “goog_lng” contains the longitude information.<br>
h. The “formatted_address” contains the standardized address based on the returned latitude and longitude.<br>
i. The “street_number” indicates the precise street number.<br>
j. The “route” indicates a named route.<br>
k. The “locality” indicates an incorporated city or town political entity.<br>
l. The “postal_code” indicates a postal code as used to address postal mail within the country.<br>
m. The “postal_code_suffix” indicates a postal code suffix.<br>
n. The “administrative_area_level_1” indicates a first-order civil entity below the country level. Within the United States, these administrative levels are states.<br>
o. The “administrative_area_level_2” indicates a second-order civil entity below the country level. Within the United States, these administrative levels are counties.<br>
p. COUNTRY contains the country information.

There are more information that have been returned by the Google map API. In this macro, it only extracts the common information. You can go to Google website to get more information about the output using this web address https://developers.google.com/maps/documentation/geocoding/intro#geocoding.

## THE GETGEOINFO MACRO CODES

Presented below are the SAS codes for the getGeoInfo macro.

```
options NOSLEEPWINDOW;
%macro getGeoInfo(libnm=,datanm=);
%macro GeoInfo(no=,address=,apikey=);
%let
url=%nrbquote(')%nrstr(https://maps.googleapis.com/maps/api/geocode/json?ad
dress=)&address.%nrstr(&key=)&apikey.%nrbquote(');
filename in url &url.;
libname in json;

data _null_;
*rc=sleep(1,0.001);
call sleep(1,0.001);
run;

proc datasets library=in;copy out=work;run;quit;

data tmp0;
```

```
set Root;
keep status;
run;

proc sql;
create table tmp1 as
select r1.ordinal_results,lat as goog_lat, lng as goog_lng,
formatted_address length=150,location_type length=30
from Geometry_location as g,results as r,Results_geometry as r1
where r1.ordinal_geometry=g.ordinal_geometry and
r1.ordinal_results=r.ordinal_results;

create table tmp2 as
select ordinal_results,ordinal_types,long_name,short_name length=100,types1
from Results_address_components as r, Address_components_types as a
where r.ordinal_address_components=a.ordinal_types
order by ordinal_results, ordinal_types;
quit;
proc transpose data=tmp2 out=tmp2(drop=_name_);by ordinal_results;id
types1;var short_name;run;

data tmp3;
merge tmp1 tmp2;
by ordinal_results;
rename ordinal_results=result_no;
run;

proc sql;
create table geores&no. as
select &no. as no, status, t.*
from tmp0, tmp3 as t
order by result_no;
quit;
proc datasets lib=work nolist;save geores: testdata;quit;run;
libname in clear;
%mend;
data testdata;
set &libnm..&datanm.;
rowno=put(_n_,z5.);
address1=tranwrd(strip(address)," ","+");
length sascodes $500.;
sascodes=cats('%GeoInfo(no=',rowno,',address=%str(',address1,'),apikey=',ap
ikey,');');
run;

data _null_;
set testdata;
call execute(sascodes);
run;

data all_geocode;
set geores:;
run;

proc sql;
create table result_geocode as
select t.address,a.*
```

```
from &libnm..&datanm. as t, all_geocode as a
where input(t.rowno,best.)=a.no
order by a.no,a.result_no;
quit;

data result_geocode;
format address status location_type no result_no goog_lat goog_lng
formatted_address street_number
route locality postal_code postal_code_suffix administrative_area_level_1
administrative_area_level_2 country;
set result_geocode;
run;
%mend;
```

## CONCLUSION

The macro presented in this paper can be used as an easy tool to get the geo coding information from the Google map API. Currently the macro only extracts the common information from the returned results. The user can update the macro to extract more information based on the user's needs.

## ACKNOWLEDGMENTS

The author wishes to thank the Division of Biostatistics and Epidemiology at Cincinnati Children's Hospital Medical Center for its support.

## CONTACT INFORMATION

Do not change the heading style or the text "CONTACT INFORMATION" of the preceding heading. Do not change the text of the following paragraph. Replace all fields that are shown in angle brackets.

Your comments and questions are valued and encouraged. Contact the author at:

Ting Sa
Division of Biostatistics and Epidemiology, Cincinnati Children's Hospital Medical Center
5136363674
Ting.Sa@CCHMC.ORG