Geocoding: Tableau and R Integration

Financial Services and Business Analytics Lab

Jose Luis Rodriguez
IT & Data Specialist

May 25, 2016
Tableau & R

Setting up R

Use Case

Tableau Geographic Roles
R Engine in Tableau

- R functions and models can now be used in Tableau by creating new calculated fields that dynamically invoke the R engine and pass values to R. The results are then returned back to Tableau for use by the Tableau visualization engine.
Who is this feature intended for?

This feature is primarily targeted for users who are already proficient in R. These users will find the integration beneficial for several reasons:

▶ Ability to do statistical analysis on Tableau data.
▶ Full access to any R package or function that has been installed in R.
▶ Use tableau visualization capabilities to further analyze and understand data without having to manipulate data in R.

Users must have access to an R server/engine to access R functions from Tableau.
Tableau & R Interaction

Access data in Tableau

1) Send Data
2) Compute
3) Receive

R Engine
Required R Packages

- **Rserve**: Rserve acts as a socket server (TCP/IP or local sockets) which allows binary requests to be sent to R.
- **ggmap**: A package for spatial visualization with Google Maps and OpenStreetMap.
R Packages Installation

First we need to open R and install the `Rserve` package to send code from tableau to R and `ggmap` to make requests to the google maps API.

```r
# Code to install the packages
install.packages(Rserve);
install.packages(ggmap);
```
Loading Packages and Starting R Serve

In the same R script after installing the R packages successfully. You need to load Rserve package with the following command.

```r
library(Rserve);
```

Starting the R Serve

After the package is loaded we need to start the Rserve with the following command.

```r
Rserve()
```
Complete Setup Code to integrate R and Tableau

This is how your R setup script would look like. In this stage we are just starting the R serve to integrate Tableau with R. It is important to make sure that the package `ggmap` to get the geocodes is installed.

```r
# This code makes sure that all the necessary packages are in the computer.
if (!require("Rserve")) install.packages("Rserve");
if (!require("ggmap")) install.packages("ggmap");

# Load the Rserve package
library(Rserve);

# Starts the Rserve
Rserve()
```

Jose Luis Rodriguez| IT & Data Specialist
Geocoding: Tableau and R Integration
Geolocation from Address

▶ In many cases we have access to data with location information ie. Address, City, County, Zip Code, Country. To visualize this data with more detail we need to use geolocation.

▶ In order to get the actual geolocation coordinates (Longitude, Latitude) for the location data, we need to calculate them.

▶ Using the R package **ggmaps** we can easily generate the geolocation coordinates
Testing the Tableau and R Connection

After starting the Rserve in R we need to test if tableau can talk to R.
Testing the Tableau and R Connection

For this case the Rserve runs on the same computer. Hence we dont have to change the default settings. We just need to test the connection, a successful connection notification window should pop-up.
Special Calculated field: Full Address

First we need to concatenate the address into a single string. To do this we need to create a calculated field.

**Calculated Field Name:** Full Address

**Tableau Formula:**

```
[Address] + "," + [City] + "," + [State]
```
Special Calculated field: Geocode function

Now that we have a complete address with the proper format. We need to create another tableau calculated field that uses R code to get the geocodes from the Google maps API.

Calculated Field Name: Geocode

Tableau Formula:

```r
SCRIPT_STR("library('ggmap');
geo <- geocode(.arg1, output = 'latlon');
geo$latlon <- do.call(paste,c(geo[c('lat','lon')], sep='','));geo$latlon",ATTR([Full Address]))
```
Special Calculated field: latitude

Calculated Field Name: **lat**

Tableau Formula:

```plaintext
left([Geocode], find([Geocode], ",")-1)
```

Now we need to change the geocodes from the **lat** calculated field to float numerical values and create a new latitude field.

Calculated Field Name: **latitude**

Tableau Formula:

```plaintext
float([lat])
```
Special Calculated field: longitude

Calculated Field Name: \texttt{lon}

Tableau Formula:

\texttt{right([Geocode], len([Geocode])-find([Geocode], ",",))}

Now we need to change the geocodes for the \texttt{lon} calculated field to float numerical values and create a new longitude field.

Calculated Field Name: \texttt{longitude}

Tableau Formula:

\texttt{float([lon])}
Assigning geographical roles

Assigning “Geographic Roles” to latitude and longitude allow us to plot latitude and longitude in Tableau as if they were in the original data set.
Setting Geographic Role for Latitude and Logitude fields
Mapping Address to Geographic coordinates

Add the calculated Latitude field to the workbook. Disregard any connection error Tableau is trying to run the R script but is missing the full address field.
Mapping Address to Geographic coordinates

Now add the Longitude field to the workbook. The connection error will appear again.
Mapping Address to Geographic coordinates

Now add the Full Address calculated field to the workbook. Tableau should start computing and mapping the addresses to the map.
Full Address to Geographical Location

File: Full Address to Geographical Location

- **Full Address**: 926 W FULTON MARKET, CHICAGO, IL
- **Latitude along Full Address**: 41.8870
- **Longitude along Full Address**: -87.6508

**Data**

- **Dimensions**:
  - Address
  - City
  - County
  - Facility Type
  - Full Address
  - Name
  - State
  - Zip Code
- **Measures**:
  - Geocode
  - lon
  - Latitude
  - Ion
  - Longitude
  - Latitude (generated)
  - Longitude (generated)
  - Number of Records
  - Measure Values

**Analytics**

- **Marks**:
  - Automatic
  - Color
  - Size
  - ABC 123
  - Label
  - Detail
  - Tooltip

**Columns**

- **Longitude**

**Rows**

- **Latitude**

**OpenStreetMap**

© OpenStreetMap contributors