

Interactive Restaurant Shiny Application

Capstone Final Project

Author: Craig Lewis

Date: November 15, 2015

Introduction

In 2015, Yelp released a series of data sets as part of the Yelp Dataset Challenge contest. These data sets included anonymized information regarding business reviews, Yelp reviewers, check-in information and tips. The objective was to make this data available to (budding) data scientists and see what can be learned about reviews and business behaviors involving of businesses reviewed on Yelp.

For this capstone project, a Shiny application was developed that allows users to work with an interactive map and see restaurants that match a users-specified selection of criteria. In developing the interactive application, the data was analyzed, cleaned and saved in advance for quick rendering and use of the map. Details of this process along with some summary metrics are provided in this paper.

Methods and Data

There is a large amount of data in the Yelp Dataset Challenge. There are five JSON files that contain different types of information regarding the businesses reviewed on Yelp. For this project, the businesses file was used. The data was read into R and then filtered to identify the categories of businesses and to then extract only the restaurants.

When the map is displayed in the Shiny application we want users to be able to select restaurants by a range of stars in the reviews as well as several other variables of interest. In order to do this, four additional variables are identified and a new dataframe is constructed using a subset of the businesses data set. The businesses data set is large. There are over 60,000 businesses in the data file and there are dozens of variables used to characterize these businesses. To simplify the Shiny application and speed the loading of the data used to display the maps, we create a new data frame with only the data of interest and then save this data off to a separate file that is loaded by the Shiny application.

Finally, a Shiny application was developed that provided the user with a map of all the restaurants that contain user specified criteria as well as the average number of stars reviewers have assigned to the restaurant. As the users, select different criteria the map is updated to show the restaurants that match that criteria.

Additional details on this process are discussed below.

Reading JSON Data

The Yelp Dataset Challenge distributes five data sets in JSON format. For the this project, only the businesses data set is used. The process was ultimately straightforward to do:

```
biz_dat <- fromJSON(sprintf("[%s]", paste(readLines(json_file), collapse=",")))
```

where **json_file** was the “**yelp_academic_dataset_business.json**” file from the Yelp data set.

Filtering Business Information

There are 61,184 business found in the businesses data set. This project is only concerned with restaurant information so in this project only the restaurants were extracted.

```
restaurants<-grep(pattern="Restaurants",biz_dat$categories)  
bars<-biz_rest<-biz_dat[restaurants,]
```

Rebuilding a Displayable Dataset

In addition to the large number of businesses found in the input file, there are a large number of columns, some of which are nested (columns within columns). While offering a rich presentation of the data, there is more in the data set than is required to map the restaurants. Moreover the structure of this data can be cumbersome. It proved easier (ultimately) to filter the data and rebuild a new data set (data frame) using only the data required in the Shiny application.

The fields extracted from the businesses data set were:

- **Business ID** - A random set of characters, presumably the anonymized name of the business. Character
- **Stars** - The number average review given by reviewers (number of stars) a number between 1 and 5. Numeric.
- **Longitude** - The geographical longitude of the restaurant location. Numeric.
- **Latitude** - The geographical latitude of the restaurant location. Numeric.
- **State** - The state (in the United States) of the restaurant. Ultimate not used. Character.
- **Take out** - Indicates whether the restaurant offers food to-go (take out food). True/False
- **Takes reservations** - Indicates whether the restaurant accepts reservations. True/False.
- **Wi-Fi** - Indicates whether the restaurant offers wi-fi. One of three values: No/Paid/Free.
- **Caters** - Indicates whether the restaurant offers catering. True/False.

This list could be readily extended to include other attributes of interest as necessary.

To simplify processing and to provide accurate information only restaurants with complete sets of information (complete cases) are filtered into the final data set to be displayed on the map.

Once the data was extracted and a new dataframe constructed, the resulting dataframe was written to a file to be read by the Shiny application. Doing this allowed the lengthy preprocessing to be bypassed and the map application to render data quickly.

The code to accomplish this:

```
bizrates<-data.frame(biz_rest$business_id, biz_rest$stars,
                    biz_rest$longitude,biz_rest$latitude,
                    biz_rest$state,
                    biz_rest$attributes$`Take-out`,biz_rest$attributes$`Takes Reservations`,
                    biz_rest$attributes$`Wi-Fi`,biz_rest$attributes$Caters)

cc<-complete.cases(bizrates)
bizrates<-bizrates[cc,]
write.table(bizrates,"bizrates.dat")
```

After these steps a file exists that can be read into the Shiny application and used to display the map.

Map Rendering with Leaflet

Leaflet is an amazing add-on for R that provides for interactive maps including within Shiny applications. Originally developed as an open-source JavaScript library it allows data to be easily overlaid on maps that can be panned and zoomed. Markers can be placed on maps and customized based on the app designer needs.

Additional information on Leaflet and R is found here: [**https://rstudio.github.io/leaflet/**](https://rstudio.github.io/leaflet/)
(<https://rstudio.github.io/leaflet/>)

Rendering a map is ultimately fairly easy to do. The code to do so is listed below:

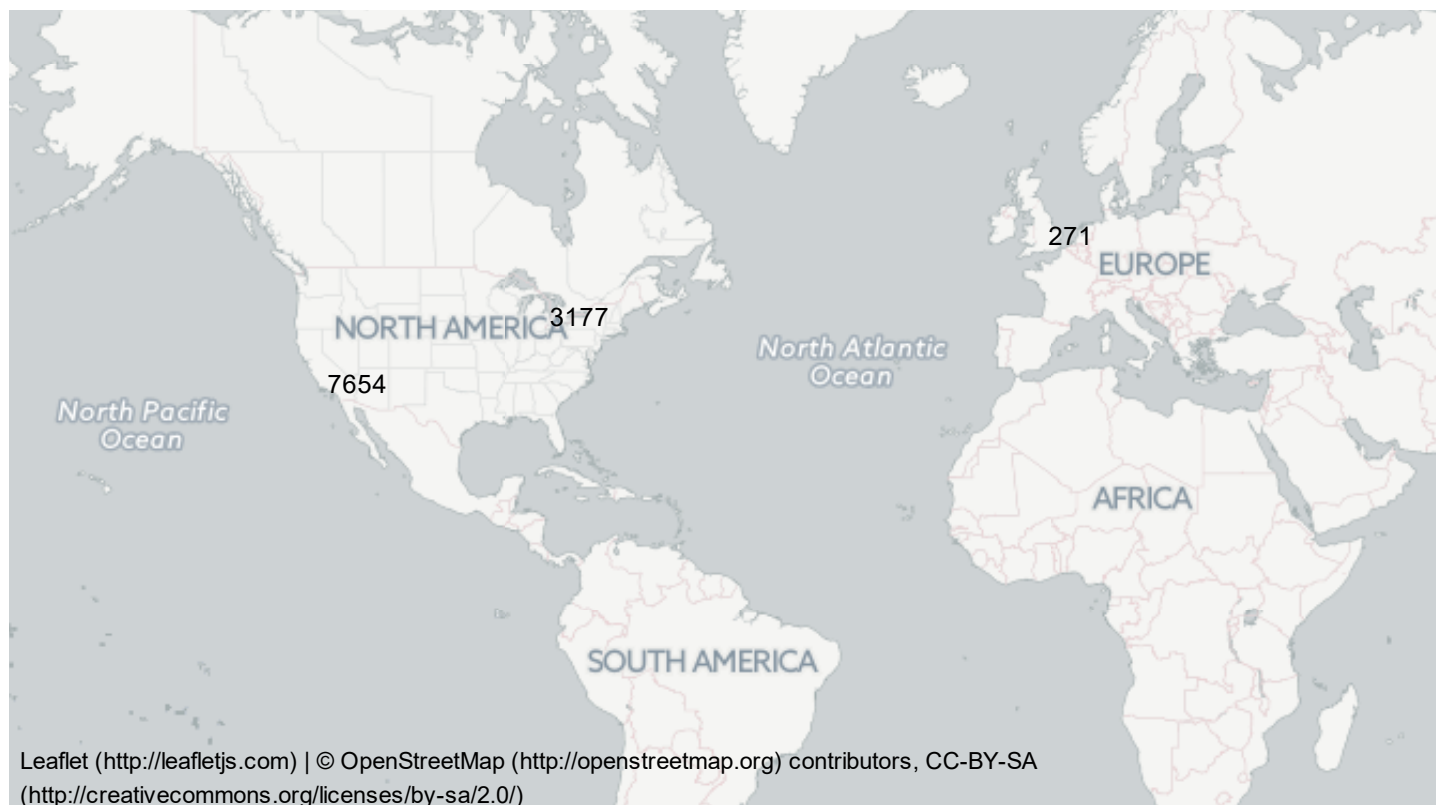
```
library(leaflet,quietly=TRUE)
library(maps,quietly=TRUE)

# Read pre-processed data
bizrates<-read.table("bizrates.dat")

# Build out a Leaflet object
mymap<-leaflet() %>%
  addTiles() %>%
  addTiles(urlTemplate = "http://{s}.basemaps.cartocdn.com/light_all/{z}/{x}/{y}.png") %>%
  mapOptions(zoomToLimits="always") %>%
  addMarkers(lat=bizrates$biz_rest.latitude,lng=bizrates$biz_rest.longitude,
            clusterOptions = markerClusterOptions(),popup=bizrates$biz_rest.business_id)

# And display it
mymap
```





Note: In a live HTML document the map works! Not so in a PDF, however.

Examining the `leaflet()` call includes the “stream” operator `%>%` to pipe output from one command to another. Of interest is the specification of `markers()` based on the latitude and longitude of the business. If there are a large number of markers in a geographical location a special cluster marker is placed indicating the number of markers contained within. Finally if a marker is clicked then we can specify the text to display, in this case we use the business ID.

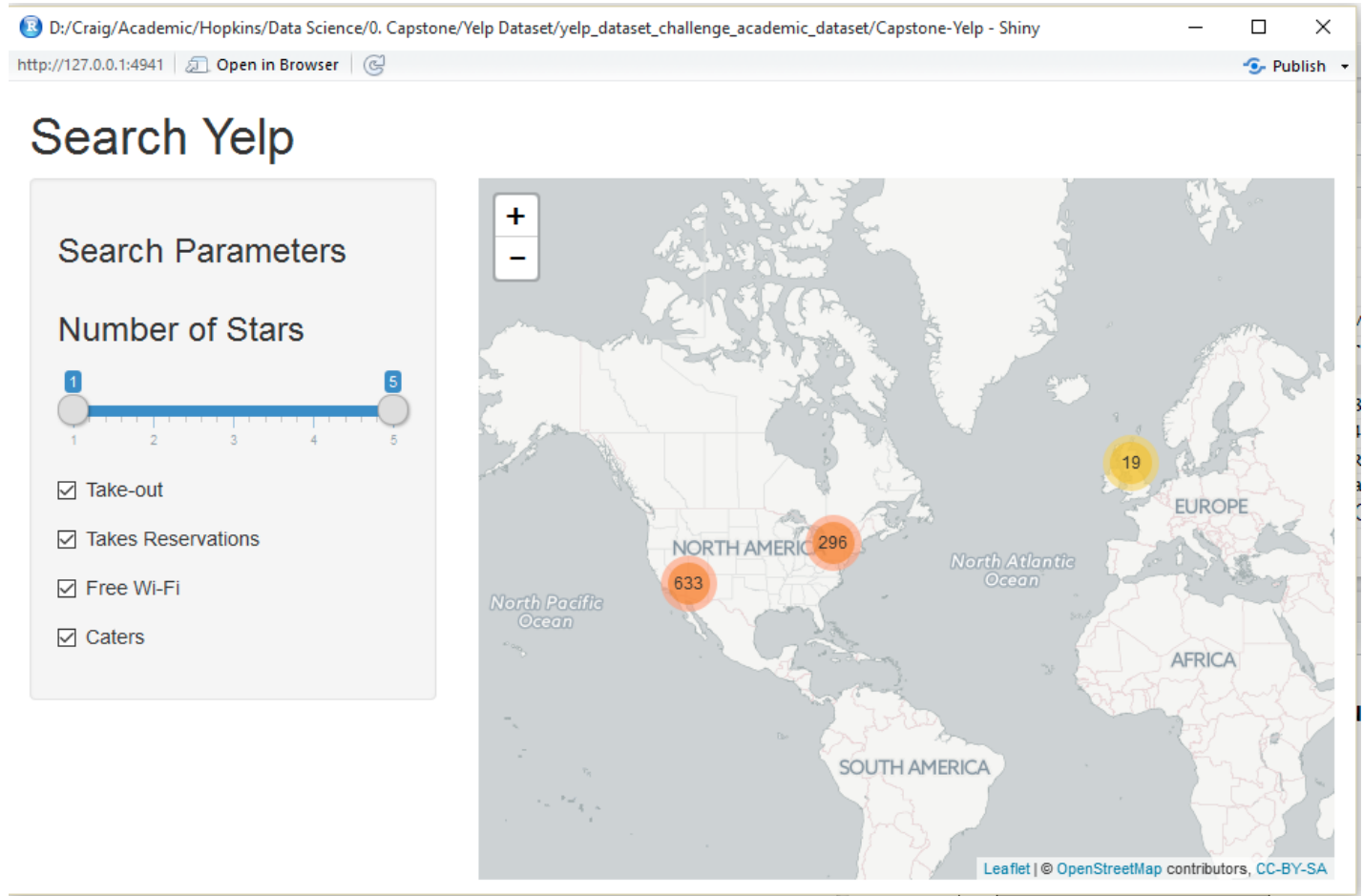
Shiny Application with User Input

The Shiny application uses the standard UI/Server configuration. The data is loaded into the app and a basic UI is presented

The user can select the range of stars in the review and then select the attributes described above: Take-out, Reservations, Free Wi-fi or Catering. The leaflet map can be zoomed or panned. Clicking on the numbers drills into the geographic region until a specific restaurant is identified. Clicking on the icon for a restaurant brings up the Business ID (which would be the name in a non-anonymized data set).

Results

Here is a screen image of the application (not functional in this rendering):



Discussion

It works! This simple Shiny application can be expanded to allow for further refinement of the search capability to provide more granular selection of restaurants. The types of businesses could be expanded to include other eating establishments. Alternatively, this structure could allow for an alternate presentation of a business category, for example, car repair as the framework exists for 1) pre-processing the data and 2) plotting the data, with selection criteria, onto a leaflet map.