Semester Project

Stephen Trippy

Data 2401 - Spring 2021

Introduction

Say hello to Yelp Fusion! This is an API with excellent documentation that allows access to almost any restaurant in the world. However, there are limits on the number of restaurants I can pull in any one request, and the search algorithm itself is a bit of a black box. Thus, getting all the data needed to answer meaningful questions becomes a puzzle, one that I'll do my best to solve over the course of this project. In cleaning and manipulating the data, I hope to answer the following question: Does the average price of restaurants in a given zip code in Harris County correlate with median income in that zip code, and can we infer why or why not?

Along the way, we'll learn more about the Yelp API, and how to access APIs in general!

Libraries! I need a few:

```
library("dplyr")
library("jsonlite")
library("httr")
library("readxl")
library("tidyr")
library("ggplot2")
library("devtools")
library("RCurl")
library("sf")

#install_github('arilamstein/choroplethrZip@v1.5.0')
library("choroplethrZip")
```

Part 1: Building the Dataset

Creating vector of Harris County Zip codes

This will allow us to query restaurants by zip code. Zip code data was copied from https://www.zillow.com /browse/homes/tx/harris-county/ and pasted into a .csv file

Before doing this, we must make sure that the csv file is in our working directory

```
harris_zips <- read.csv('harris_zipcodes.csv')
```

Reading in Census Data

Next, we read in census data, which is taken from 2006-2010. While unfortunately the api data is current, we can make the assumption that wealth levels per zip code have not changed too much

An added benefit of using census data is that this will give us a list of *only valid* zip codes. *i.e. when we query, we will only be using valid zip codes that won't break the process*

Census data was sourced from: https://www.psc.isr.umich.edu/dis/census/Features/tract2zip/

```
#read the demographic data in. cells with no data are marked with a '.'

zip_demographic_data <- read_excel('zipcode_census_data.xlsx', sheet = 'nation', na = ".") %>%
  rename(zip_code = Zip)
harris_data_by_zip <- inner_join(harris_zips, zip_demographic_data, by = 'zip_code')

#Some of the zip codes in harris_zips had no demographic data, so i decided to
#inner_join.
#I could left join though, if I really felt like keeping all of the restaurant data
#from these zip codes
```

Let's play around with the data we've just gotten

```
harris_data_by_zip %>%
  arrange(desc(Median)) %>%
  head(10)
```

```
##    zip_code    Median     Mean   Pop
## 1     77401 153648.53 204877.7 17491
## 2     77094 148245.14 170986.5  8498
## 3     77005 141928.87 209889.7 23151
## 4     77059 133259.02 151170.6 16949
## 5     77024 132759.43 210545.4 34775
## 6     77345 123437.99 154831.6 26122
## 7     77079 102246.16 131198.1 31280
## 8     77450 101498.39 121174.3 71889
## 9     77379 100527.40 120940.0 70544
## 10    77546  95641.87 109594.8 47636
```

```
#small sample of the 10 wealthiest zip codes by median income in Harris county.
#Neat!
```

Preparing the URI

We now prepare the address we'll be using to query. I have a personal API key, which is stored in a separate file. If you wish to reproduce this code, you'll need to file a Fusion application and provide your own "yelp_api_key.R" file

The format of this file should be:

client_ID <- "Your_Application_ID"

yelp_key <- "Your_Yelp_Key"

```
#Loading the API key from a separate file
source("yelp_api_key.R") #allows yelp_key to be available
#info in order to make GET request, using API key as a header
base_uri <- "https://api.yelp.com/v3"
endpoint <- "/businesses/search"
search_uri <- paste0(base_uri, endpoint)
```

Unleashing the Beast

Now, we prepare to query. My plan is to cycle through every zip code in Harris county, pulling as many restaurants as there are available from each zip code.

There are, however, 2 problems with this approach:

- 1. The yelp api doesn't return data the way I'd like it to. When I tried to query by one particular zip code, restaurants from other zip codes would get pulled back. I attribute this to the fact if you are located in a certain zip code, yelp would want to show you nearby restaurants from different zip codes
  - Solution: I figure that if I pull enough restaurants, the overlap between zip code queries will be enough to make a representative sample. **Note: This is an inherently flawed assumption, but I will be using it regardless to compile my dataset**
- 2. Problems with my nested 'for loop' breaking: Initially, my plan was to cycle through each zip code until I compiled what I thought would be a representative sample of restaurants. However, my loops put me over the max requests, and I was only able to get a one-time max of 50 results per zip code.
  - Solution: I'm just going to have to work with the limit I have. Unfortunately, results may be excluded based on whatever parameters the yelp database decides merits a return. Nonetheless, I believe I may still get a representative enough sample to answer my questions

```
zip_restaurant_list = list()
restaurant_list = list()

#the following loop iterates through the list of valid zip codes
#and sets the location query equal to each zip code in the list
for(j in 1:length(harris_data_by_zip$zip_code)){
  zip_code_n <- harris_data_by_zip$zip_code[j]

    query_params <- list(
      term = "Restaurants",
      location = zip_code_n, #zip code at position 'j' in the dataframe
      sort_by = "distance",
      limit = 50
    )
    response <- GET(
      search_uri,
      query = query_params,
      add_headers(Authorization = paste("bearer", yelp_key))
    )
    response_text <- content(response, type = "text")
    response_data <- fromJSON(response_text)
    restaurants <- flatten(response_data$businesses)

    #each set of restaurants pulled is added to the restaurant list
    restaurant_list[[j]] <- restaurants
}
```

Compiling our data

We have our restaurant data! However, it's spread out across about 134 lists, with a fair bit of overlapping data. The next step is to combine all these lists into one dataset.

```
#take all of the data we've just extracted. Now bind it into one masterlist
masterlist <- bind_rows(restaurant_list)

#remove the duplicate restaurants (each restaurant should have a unique id)
masterlist <- masterlist[!duplicated(masterlist$id), ]

#renaming the zip code column will make it so that we can join with the
#harris_data_by_zip dataset later. We also set this variable's type to double
#for the same reason
masterlist <- masterlist %>% rename(zip_code = location.zip_code)
masterlist$zip_code <- as.double(masterlist$zip_code)
```

So how many restaurants were we able to pull into this dataframe?

```
nrow(masterlist)
```

```
## [1] 5638
```

```
#not too bad!
```

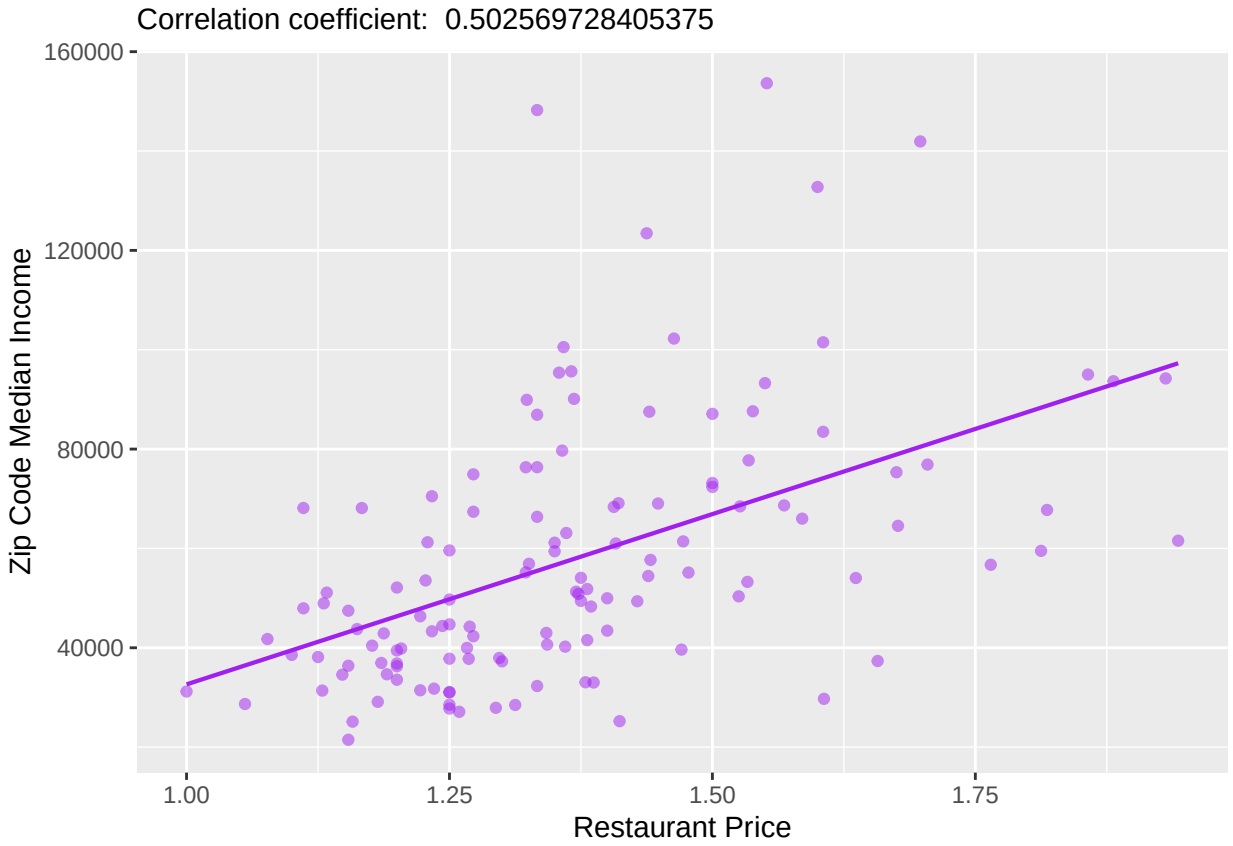
This ugly bit of code is the final step of data cleaning. We turn the price variable into data we can work with

```
masterlist$price[masterlist$price == "$"] <- 1
masterlist$price[masterlist$price == "$$"] <- 2
masterlist$price[masterlist$price == "$$$"] <- 3
masterlist$price[masterlist$price == "$$$$"] <- 4
masterlist$price <- as.double(masterlist$price)
```

Part 2: Answering the question

Does the average price of restaurants in a given zip code correlate with median income in that zip code? Can we infer why or why not?

```
zip_data_sublist <- masterlist %>%
  group_by(zip_code) %>%
  filter(n() >= 10) %>% #10 was chosen somewhat arbitrarily, but excludes zip codes
  #with too few restaurants in them
  summarize(restaurant_price = mean(price, na.rm = TRUE)) %>%
  right_join(harris_data_by_zip, by = "zip_code")

#find the correlation coefficient between restaurant price and median income
median_price_coeff <- cor(zip_data_sublist$restaurant_price, zip_data_sublist$Median,
    use = "pairwise.complete.obs")
#The 'use' command here omits NA results, I believe)

zip_data_sublist %>%
  ggplot(aes(restaurant_price, y = Median)) +
  geom_point(alpha = 0.5, color = "purple") +
  labs(x = "Restaurant Price", y = "Zip Code Median Income",
       subtitle = paste("Correlation coefficient: ", median_price_coeff)) +
  geom_smooth(method = 'lm', lwd = 0.75, se = FALSE, color = "purple")
```

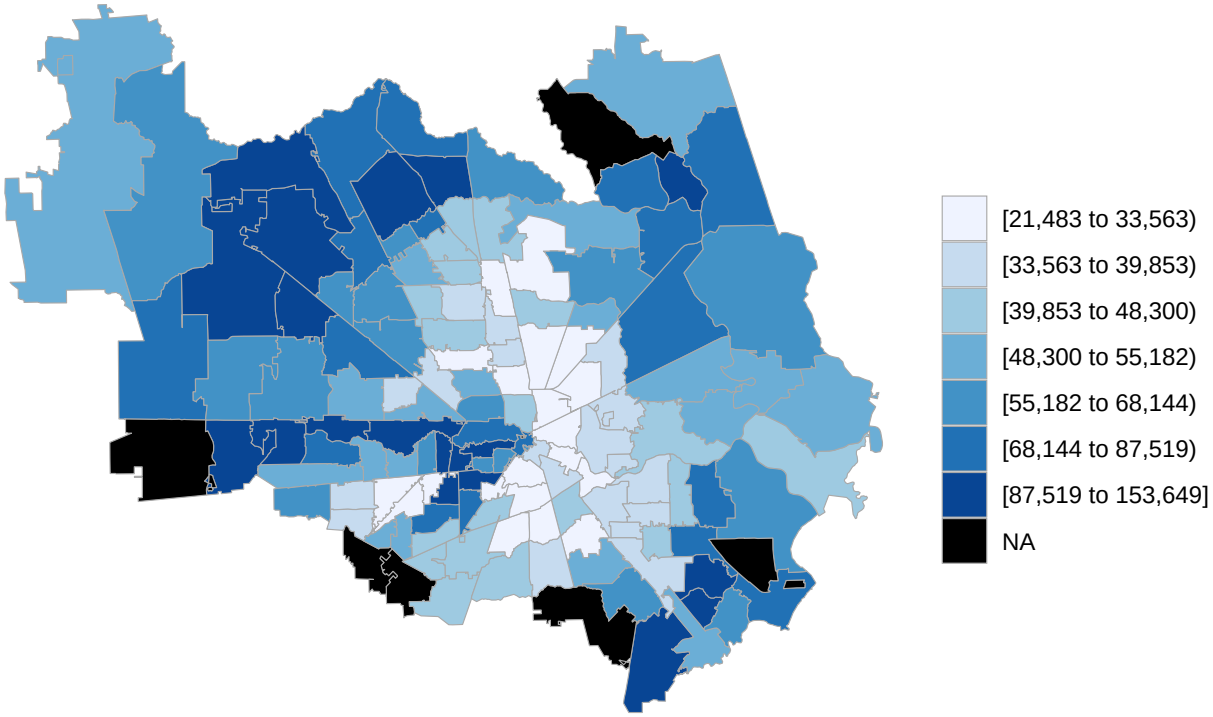


So with a correlation of right around 0.5, we are looking at a moderately strong correlation. I'm going to conclude that my theory holds!

I want to show this one other way: by mapping median income on one map and median restaurant price on another. To do this, I'll be using the choroplethrZip package.

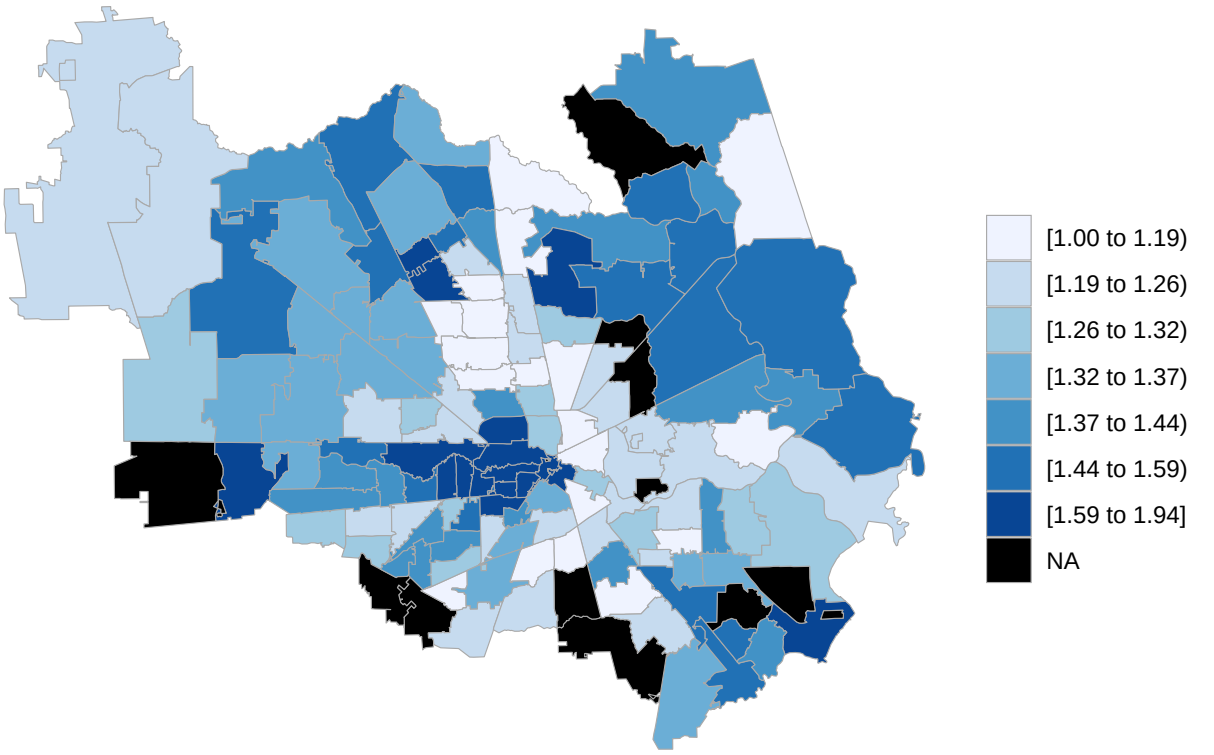
```
harris_choropleth_data <- harris_data_by_zip #create a new dataframe for this test
harris_choropleth_data$zip_code <- as.character(harris_choropleth_data$zip_code)

#Choroplethr is a bit particular about the format the dataframe argument needs to be
#in, so we have to do some cleaning:
harris_choropleth_data <- harris_choropleth_data %>%
  rename(region = zip_code, value = Median) %>%
  select(region, value)

#Upon googling we find Harris County's unique FIPS code: 48201
zip_choropleth(harris_choropleth_data,
               county_zoom = 48201,
               title = "Median Income by Zip (Dollars/yr)")
```

Median Income by Zip (Dollars/yr)




```
harris_choropleth_data <- zip_data_sublist #overwrite the previous dataframe
harris_choropleth_data$zip_code <- as.character(harris_choropleth_data$zip_code)
harris_choropleth_data <- harris_choropleth_data %>%
  rename(region = zip_code, value = restaurant_price) %>%
  select(region, value)
zip_choropleth(harris_choropleth_data,
               county_zoom = 48201,
               title = "Average Restaurant Price  by Zip (1-4 '$')")
```

Average Restaurant Price by Zip (1−4 '$')



Visually, these two maps roughly reflect a correlation in median wealth and restaurant price per zip code. Not too surprisingly, restaurants tend to be more expensive in the middle of the city (Many upscale restaurants depend on corporate lunches and dinners for their business).

We can see a crescent around downtown Houston of lower-income zip codes, and this is reflected by lower average restaurant price.

We can also see that to the west, northwest, and southeast, there are pockets of higher-income zip codes, and this is reflected by higher average restaurant price.

Final Thoughts

One thing to note is that by using an API for our data, the numbers used will update in real time! For example, the correlation coefficient actually shifted a bit between queries I ran over a couple of days.

As previously mentioned, there are a few problems with the analysis that I did. For one, I have no way of knowing how Yelp selects restaurants to show me, and while almost 6000 restaurants is a lot, it's nowhere near the total amount in Harris County. Also, I believe that certain areas inside the city may be misrepresented, as there are far more restaurants than could have been pulled.

There are also so many questions that I wanted to ask about this data but didn't have time for, and will just have to wait until another project.

Overall however, I'm incredibly pleased with how this project turned out, and I hope this has proven interesting and insightful.