Semester Project Cook Version

Stephen Trippy

6/12/2021

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 Cook County Zip codes

read in cook zip codes

```
cook_zips <- read.csv('cook_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)
cook_data_by_zip <- inner_join(cook_zips, zip_demographic_data, by = 'zip_code')

#Some of the zip codes in cook_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

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

```
##    zip_code   Median     Mean   Pop
## 1     60043 223106.2 342071.5  2513
## 2     60022 201353.6 296519.4  8153
## 3     60093 174450.5 270926.6 19570
## 4     60521 171584.8 240711.8 17597
## 5     60029 157708.0 203699.0   482
## 6     60091 137738.9 196711.4 27020
## 7     60558 128869.3 164767.7 12960
## 8     60010 117974.2 162539.2 43960
## 9     60305 112815.3 174742.0 11172
## 10    60203 110925.0 148930.1  4523
```

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

Preparing the URI

```
#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

```
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(cook_data_by_zip$zip_code)){
  zip_code_n <- cook_data_by_zip$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] 6647
```

```
#not too bad!
```

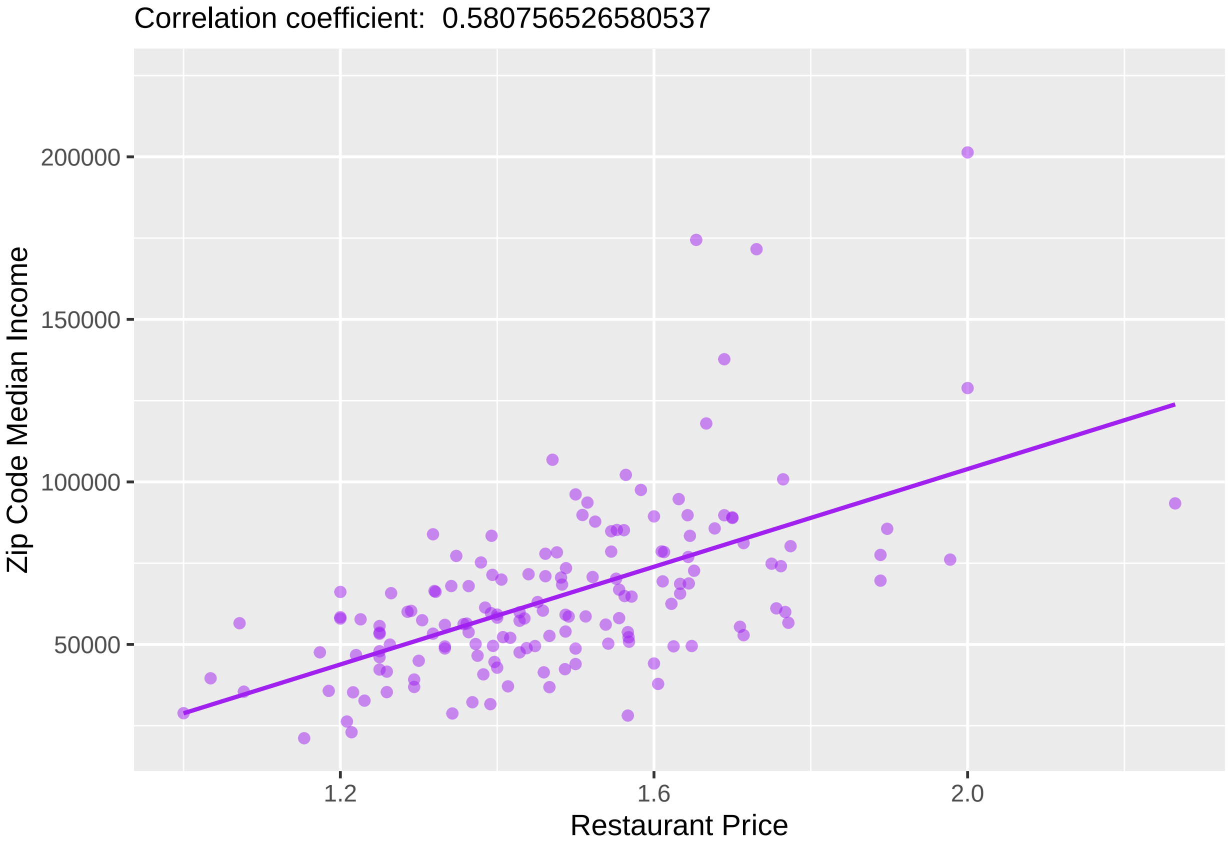
This ugly bit of code is the final step of data cleaning. We turn the price variable into data we can work with

```
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

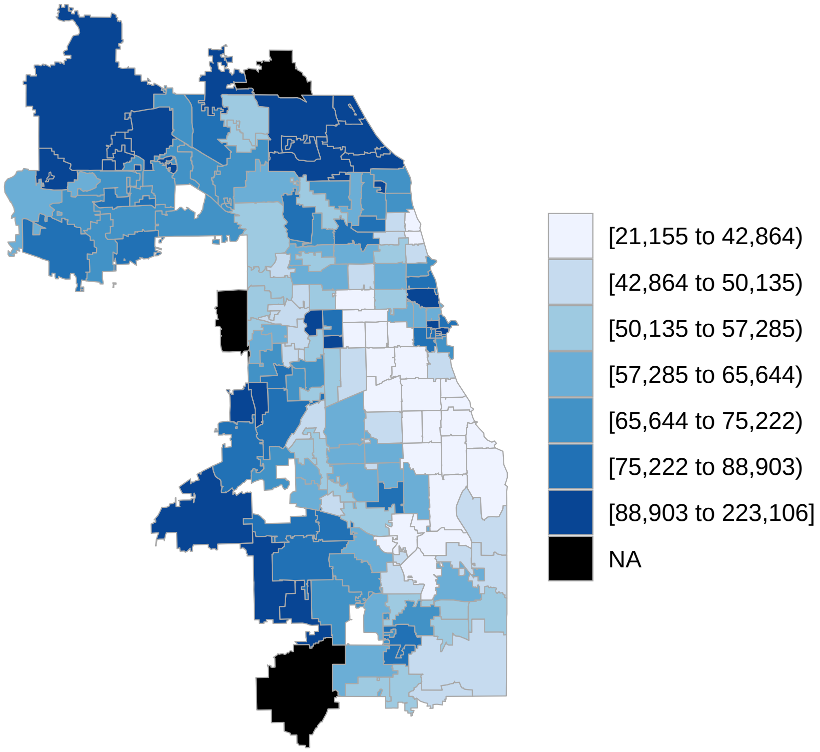
Does the average price of restaurants in a given zip code correlate with median income in that zip code? Can we infer why or why not?

```
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(cook_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")
```



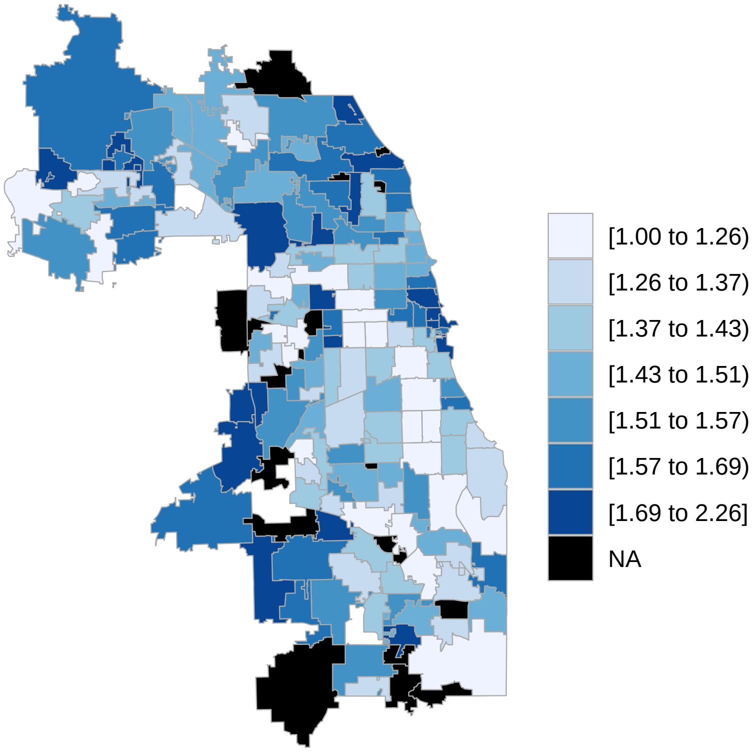
```
cook_choropleth_data <- cook_data_by_zip #create a new dataframe for this test
cook_choropleth_data$zip_code <- as.character(cook_choropleth_data$zip_code)

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

```
#Upon googling we find Cook County's unique FIPS code: 17031
zip_choropleth(cook_choropleth_data,
               county_zoom = 17031,
               title = "Median Income by Zip (Dollars/yr)")
```



Median Income by Zip (Dollars/yr)

```
cook_choropleth_data <- zip_data_sublist #overwrite the previous dataframe
cook_choropleth_data$zip_code <- as.character(cook_choropleth_data$zip_code)
cook_choropleth_data <- cook_choropleth_data %>%
  rename(region = zip_code, value = restaurant_price) %>%
  select(region, value)
zip_choropleth(cook_choropleth_data,
               county_zoom = 17031,
               title = "Average Restaurant Price  by Zip (1-4 '$')")
```



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