Lab 7 Part 1
Webscraping using R

MIT 11.188/11.520

April 8, 2020

Rida Qadri

AIMS OF LAB 7

Use R and APIs to scrape data from a website

Use Google location data to understand mobility patterns

Learn how to merge various datasets to analyze an urban problem or phenomenon

Consider the value and limitations of using digital trace data not created for the express purpose of research

LAB 7 PROBLEM STATEMENT

Social Distancing→Reduced Mobility Options→Reduced Food Options

What kind of access people have to restaurants in different areas?

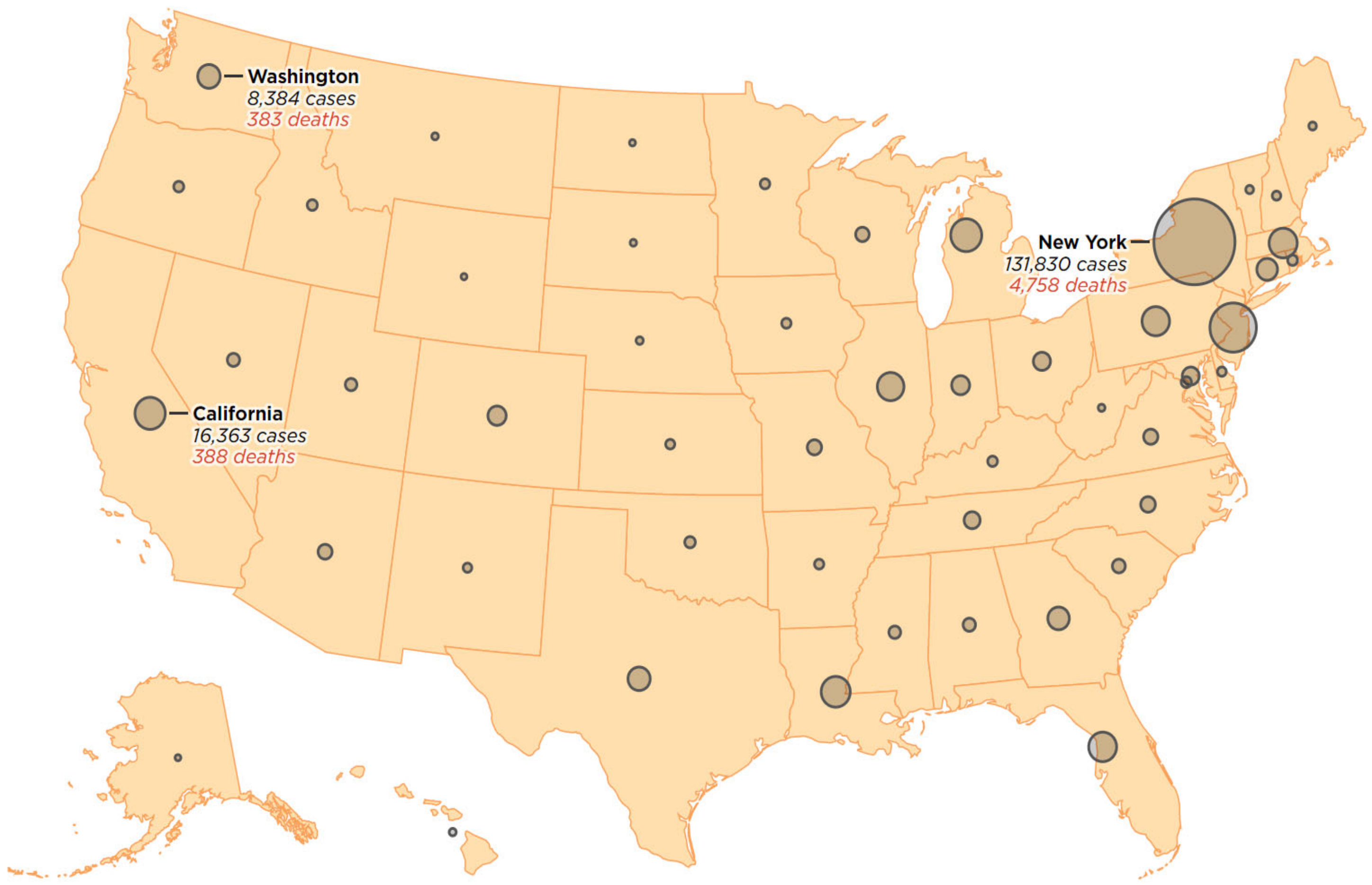
LAB 7 PROBLEM STATEMENT

Understand the food choices of one individual during lockdown, who is currently practicing social distancing in Central Square since March 12, 2020.

Part 1: Scraping Yelp Data using R

Part 2: Exploring Google Maps Location Data

Data as of 10:46 a.m. on April 7, 2020

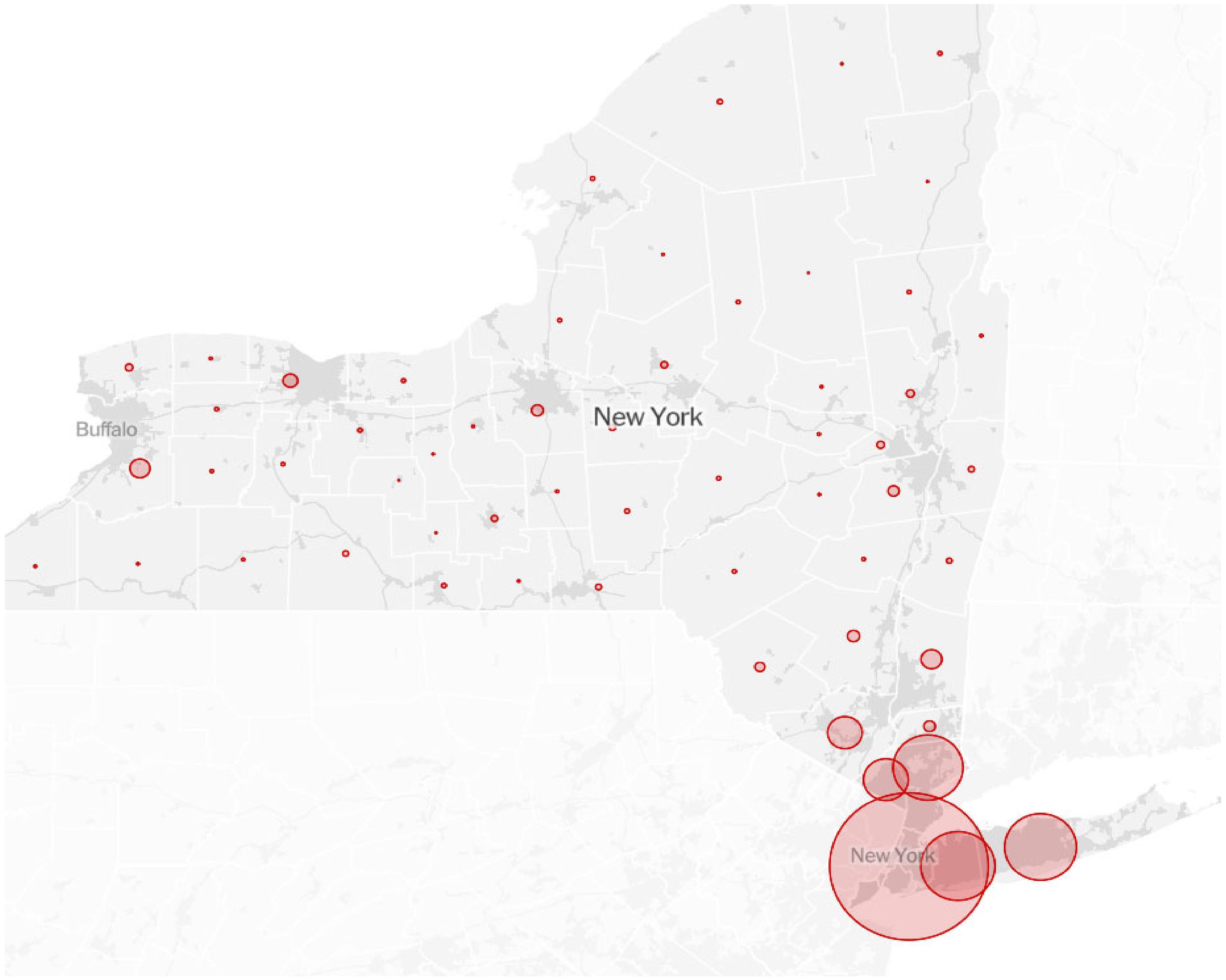


Source: Center for Systems Science and Engineering at Johns Hopkins University. Cases on cruise ships are not included.

Credit: Daniel Wood, Stephanie Adeline, Sean McMinn, Thomas Wilburn and Connie Hanzhang Jin/NPR

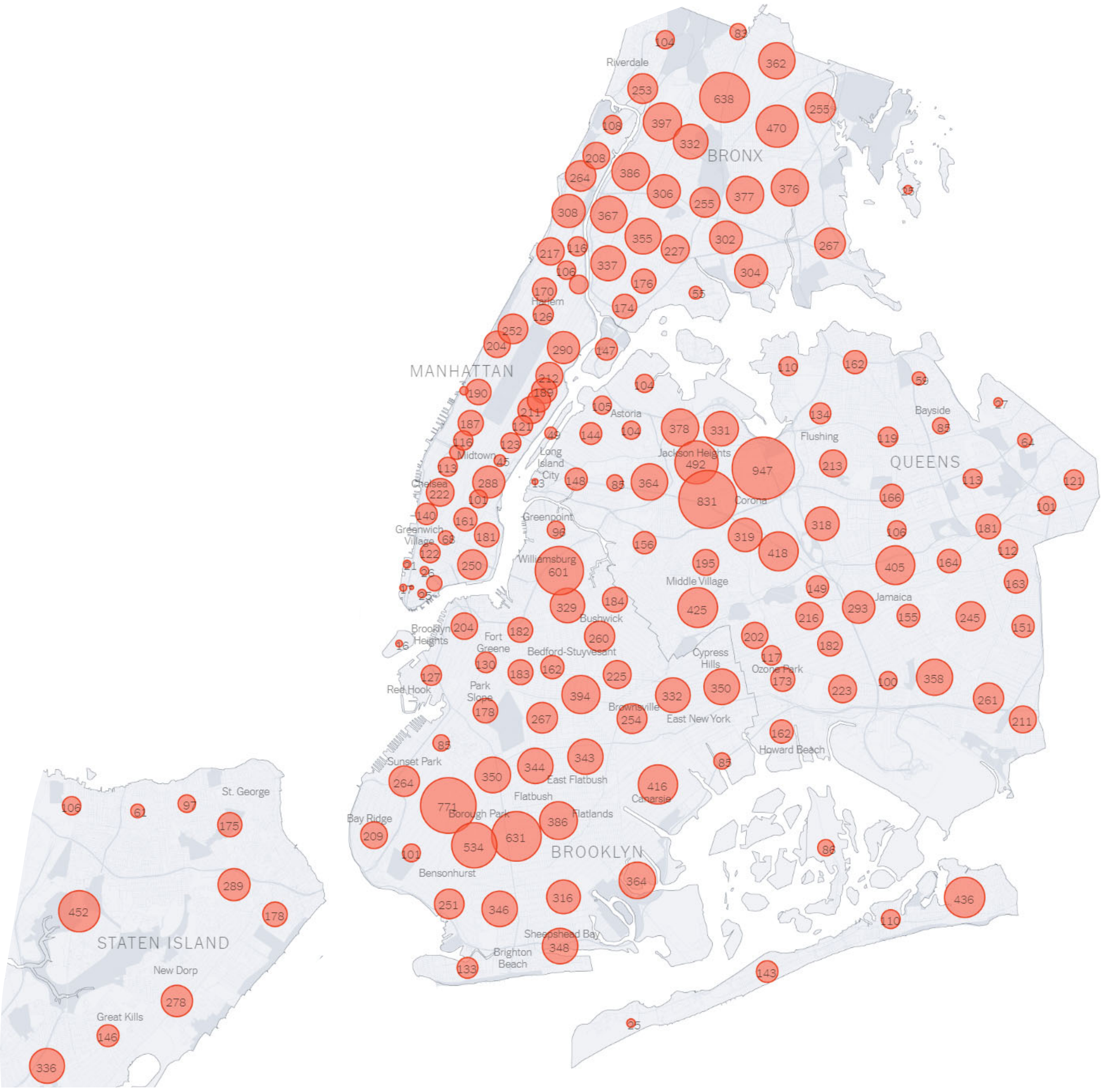
COVID-19 CASES BY STATE

Source: NPR



COVID-19 CASES BY COUNTY

Source: New York Times



COVID-19 CASES BY ZIPCODE

Source: New York Times

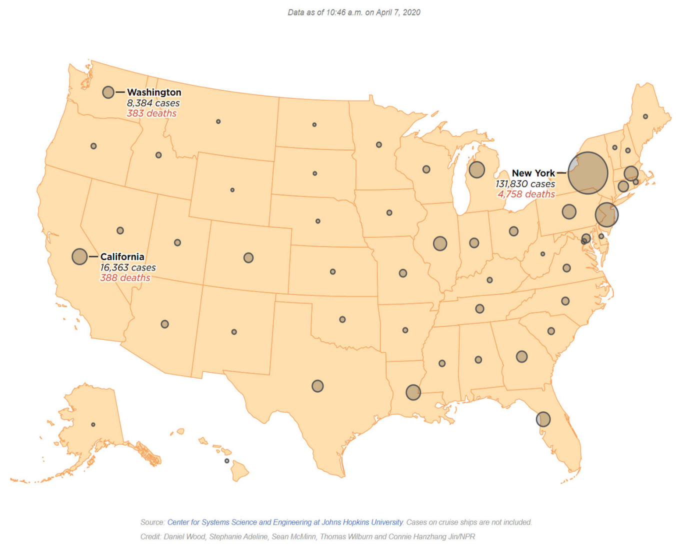
ADVANTAGES OF MORE GRANULAR DATA

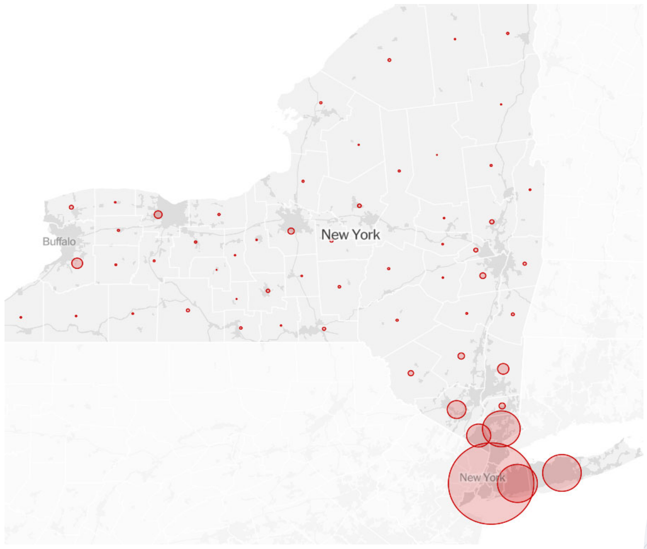
Localized behavior can get lost in premade aggregations

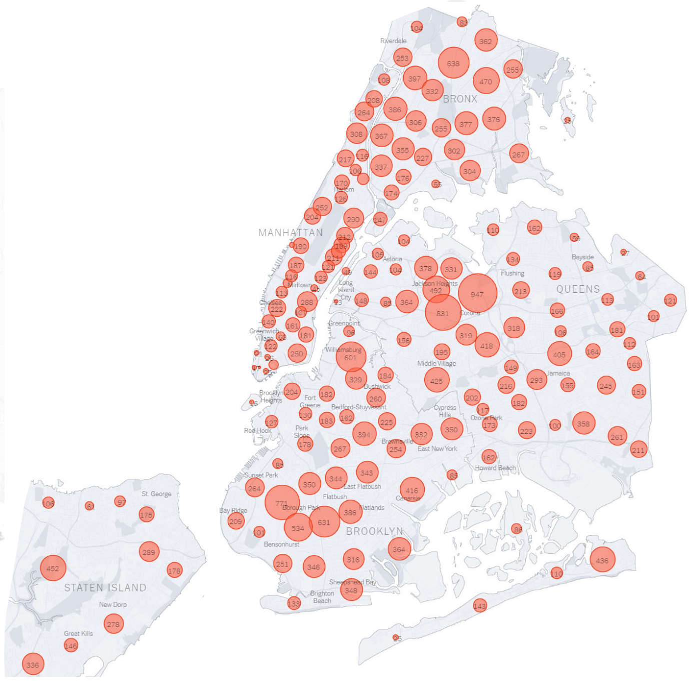
Urban patterns often do not conform to administrative boundaries

New geographies of research open up/ not reliant on existing datasets

WHAT CAN YOU TELL FROM EACH MAP?


Data as of 10:46 a.m. on April 7, 2020
Washington
8,384 cases
383 deaths
New York
131,830 cases
4,758 deaths
California
16,363 cases
388 deaths
Source: Center for Systems Science and Engineering at Johns Hopkins University. Cases on cruise ships are not included.
Credit: Daniel Wood, Stephanie Adeline, Sean McMinn, Thomas Wilburn and Connie Hanzhang Jin/NPR





Which ‘granularity’/scale is useful for what type of question?
What normalization would or would not be useful?
What other information could be provided on the map to bring to light new dimensions of the phenomenon?

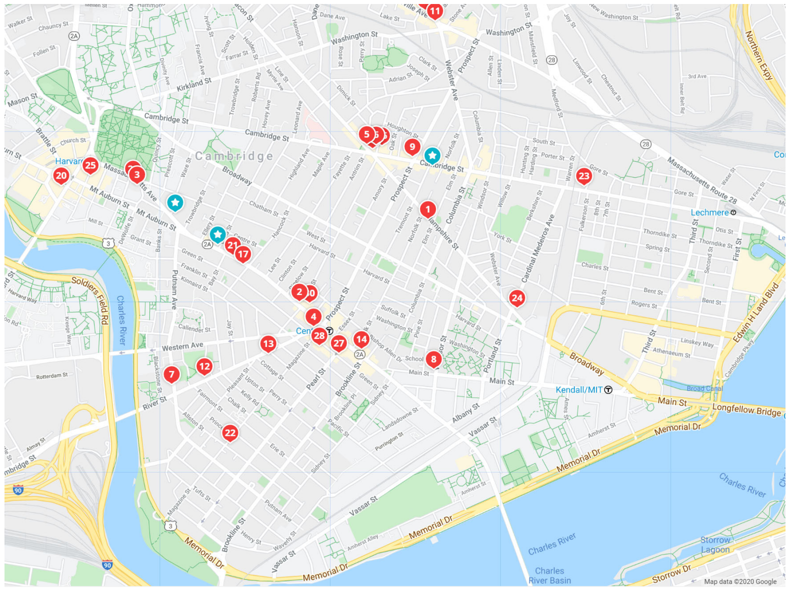
PROBLEM STATEMENT

Social Distancing→Reduced Mobility Options→Reduced Food Options

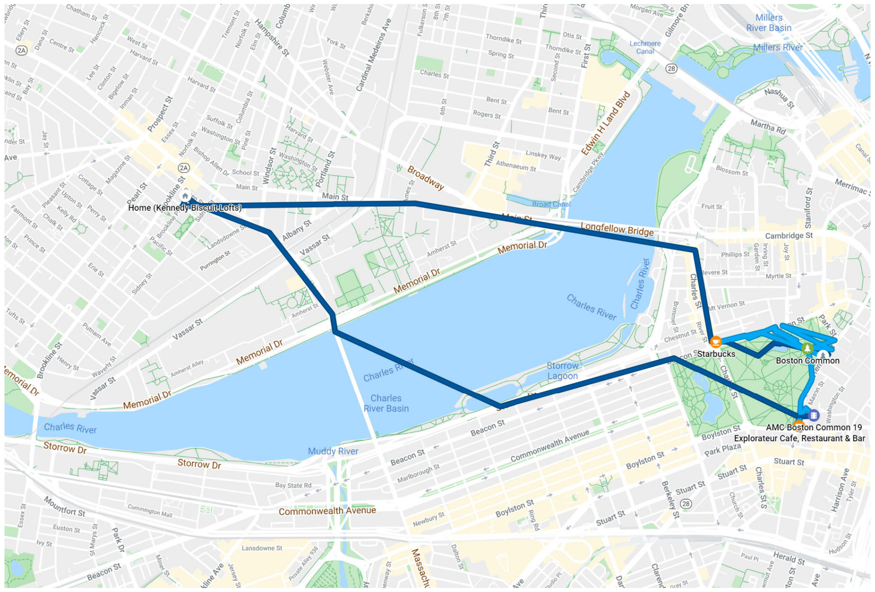
What kind of access people have to restaurants in different areas?

What kind of data would we need to answer this question?

WE ALREADY SEE THIS DATA EVERYDAY



Restaurants in an area


Home (Kennedy Biscuit Lofts)
Starbucks
Boston Common
AMC Boston Common 19
Explorateur Cafe, Restaurant & Bar

Mobility patterns recorded by Google using GPS in cell phones

HOW DO WE ACCESS THIS DATA?

Webscraping

APIs

Downloading data platforms make available

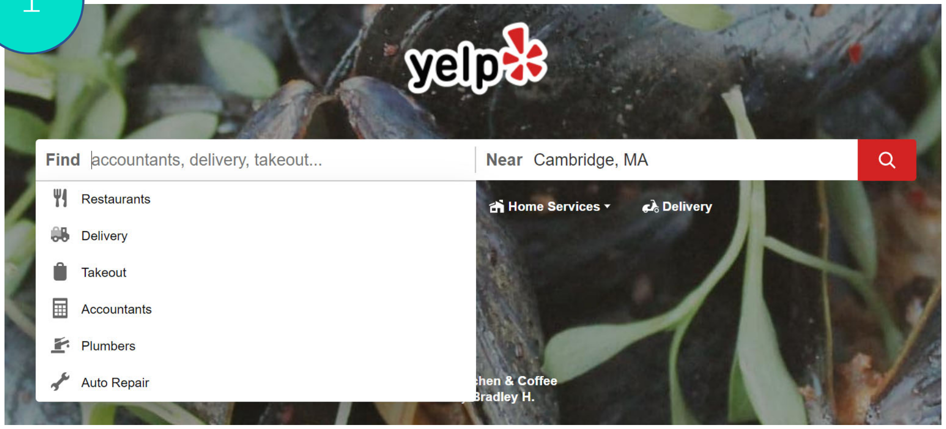
WEBSCRAPING

Information is available online, but not in publicly accessible datasets or in easily analyzable formats.

Web scraping allows you to extract information and from websites and store it in a useable way

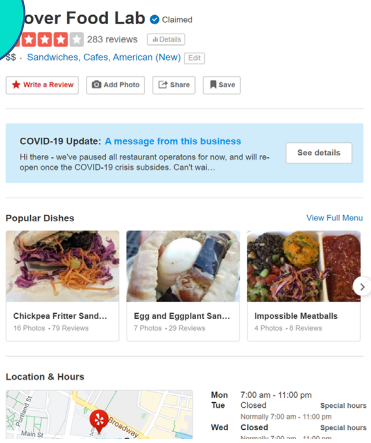
In some cases, website owners provide application programming interfaces (APIs) that establish a protocol for requesting batches of information from the website

1



Search using parameters you define

2



over Food Lab Claimed

283 reviews Details

$$ · Sandwiches, Cafes, American (New) Edit

Write a Review | Add Photo | Share | Save

COVID-19 Update: A message from this business

Hi there - we've paused all restaurant operatons for now, and will re-open once the COVID-19 crisis subsides. Can't wai...

See details

Popular Dishes — View Full Menu

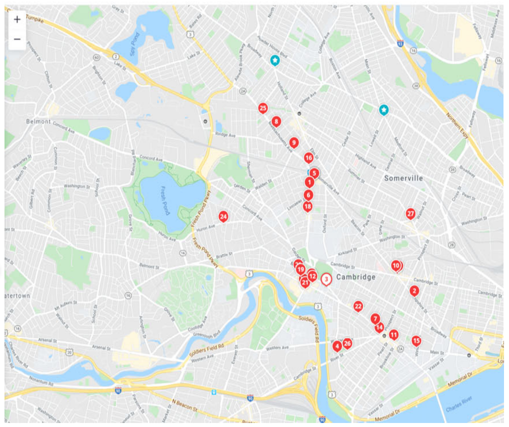
Chickpea Fritter Sand... 16 Photos · 79 Reviews

Egg and Eggplant San... 7 Photos · 29 Reviews

Impossible Meatballs 4 Photos · 8 Reviews

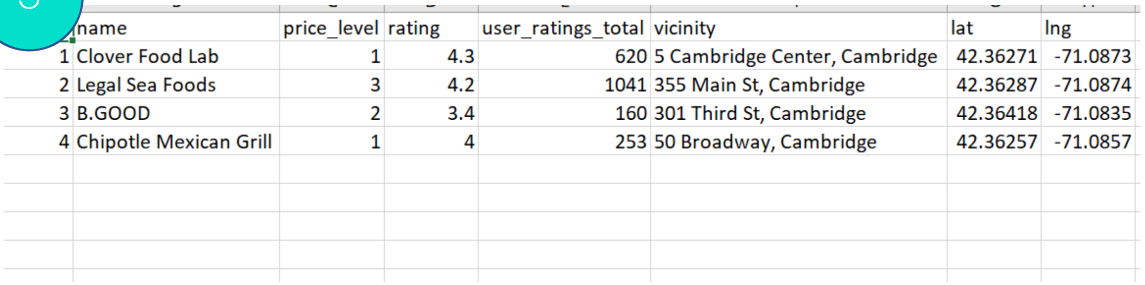
Location & Hours

Mon	7:00 am - 11:00 pm	
Tue	Closed Normally 7:00 am - 11:00 pm	Special hours
Wed	Closed	Special hours



Open each result and 'get' needed information

3



	name	price_level	rating	user_ratings_total	vicinity	lat	lng
1	Clover Food Lab	1	4.3	620	5 Cambridge Center, Cambridge	42.36271	-71.0873
2	Legal Sea Foods	3	4.2	1041	355 Main St, Cambridge	42.36287	-71.0874
3	B.GOOD	2	3.4	160	301 Third St, Cambridge	42.36418	-71.0835
4	Chipotle Mexican Grill	1	4	253	50 Broadway, Cambridge	42.36257	-71.0857

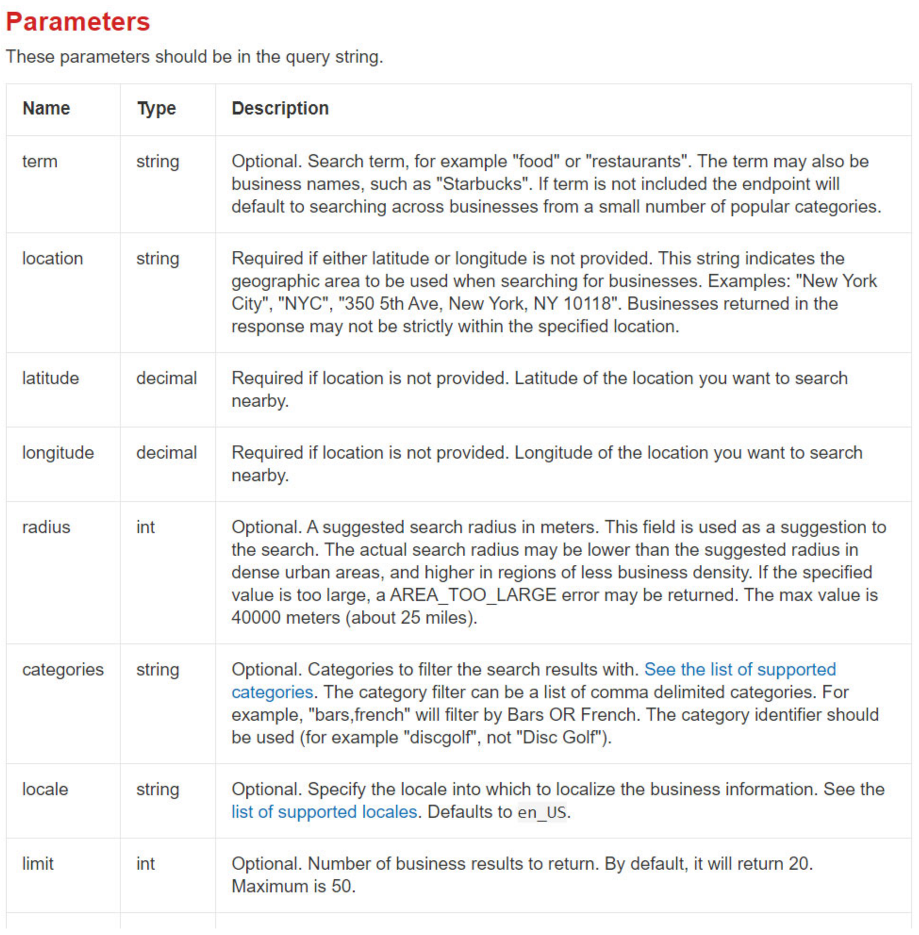
Put information in a formatted table

ELEMENTS OF WEBSCRAPING

- Script that automates
  - Doing a search using parameters you define
  - Copying all the results
  - Putting them in a formatted table for you to analyze.

- API [Application Programming Interface]:
  - Think of its as a set of operations that decide what happens when you ‘request’ something from the server
  - A webscraping API lets you have pre-defined access to a servers stored data

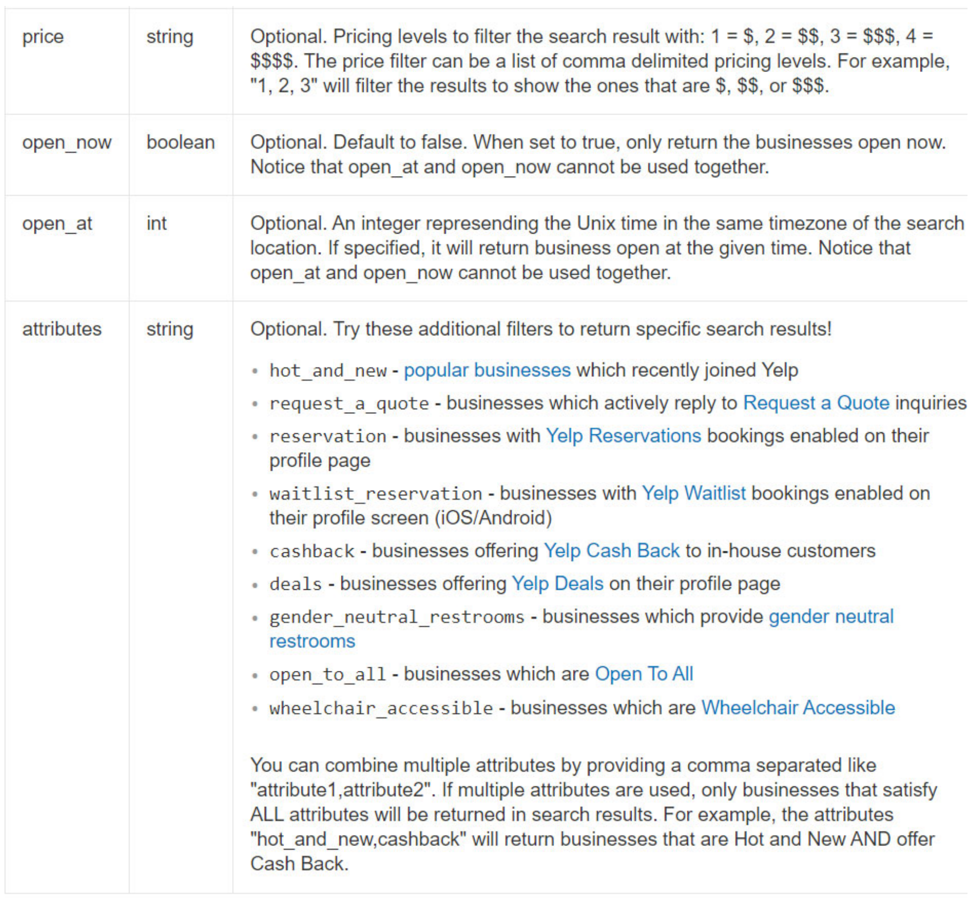
Yelp API—Search Parameters



Parameters

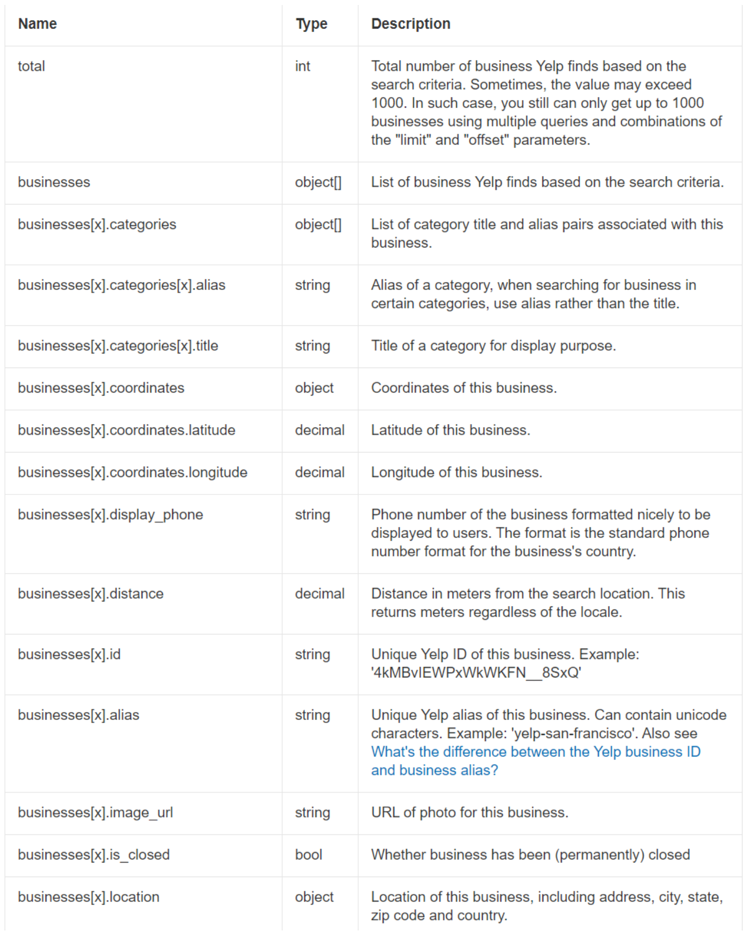
These parameters should be in the query string.

Name	Type	Description
term	string	Optional. Search term, for example "food" or "restaurants". The term may also be business names, such as "Starbucks". If term is not included the endpoint will default to searching across businesses from a small number of popular categories.
location	string	Required if either latitude or longitude is not provided. This string indicates the geographic area to be used when searching for businesses. Examples: "New York City", "NYC", "350 5th Ave, New York, NY 10118". Businesses returned in the response may not be strictly within the specified location.
latitude	decimal	Required if location is not provided. Latitude of the location you want to search nearby.
longitude	decimal	Required if location is not provided. Longitude of the location you want to search nearby.
radius	int	Optional. A suggested search radius in meters. This field is used as a suggestion to the search. The actual search radius may be lower than the suggested radius in dense urban areas, and higher in regions of less business density. If the specified value is too large, a AREA_TOO_LARGE error may be returned. The max value is 40000 meters (about 25 miles).
categories	string	Optional. Categories to filter the search results with. See the list of supported categories. The category filter can be a list of comma delimited categories. For example, "bars,french" will filter by Bars OR French. The category identifier should be used (for example "discgolf", not "Disc Golf").
locale	string	Optional. Specify the locale into which to localize the business information. See the list of supported locales. Defaults to `en_US`.
limit	int	Optional. Number of business results to return. By default, it will return 20. Maximum is 50.

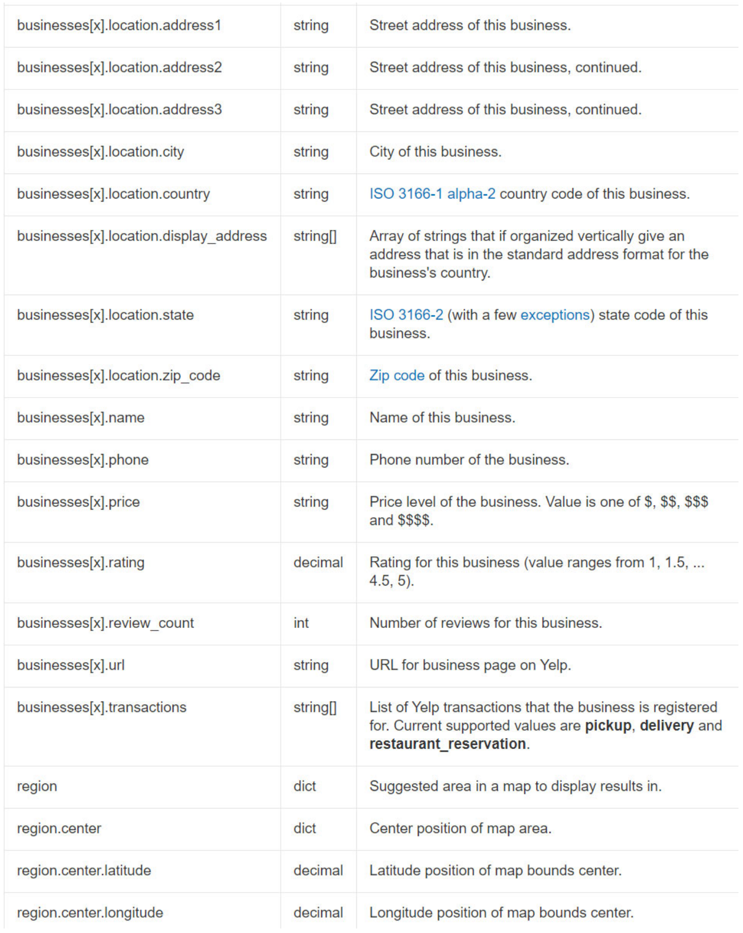


price	string	Optional. Pricing levels to filter the search result with: 1 = $, 2 = $$, 3 = $$$, 4 = $$$$. The price filter can be a list of comma delimited pricing levels. For example, "1, 2, 3" will filter the results to show the ones that are $, $$, or $$$.
open_now	boolean	Optional. Default to false. When set to true, only return the businesses open now. Notice that open_at and open_now cannot be used together.
open_at	int	Optional. An integer representing the Unix time in the same timezone of the search location. If specified, it will return business open at the given time. Notice that open_at and open_now cannot be used together.
attributes	string	Optional. Try these additional filters to return specific search results! • `hot_and_new` - popular businesses which recently joined Yelp • `request_a_quote` - businesses which actively reply to Request a Quote inquiries • `reservation` - businesses with Yelp Reservations bookings enabled on their profile page • `waitlist_reservation` - businesses with Yelp Waitlist bookings enabled on their profile screen (iOS/Android) • `cashback` - businesses offering Yelp Cash Back to in-house customers • `deals` - businesses offering Yelp Deals on their profile page • `gender_neutral_restrooms` - businesses which provide gender neutral restrooms • `open_to_all` - businesses which are Open To All • `wheelchair_accessible` - businesses which are Wheelchair Accessible You can combine multiple attributes by providing a comma separated like "attribute1,attribute2". If multiple attributes are used, only businesses that satisfy ALL attributes will be returned in search results. For example, the attributes "hot_and_new,cashback" will return businesses that are Hot and New AND offer Cash Back.

YELP API—Results



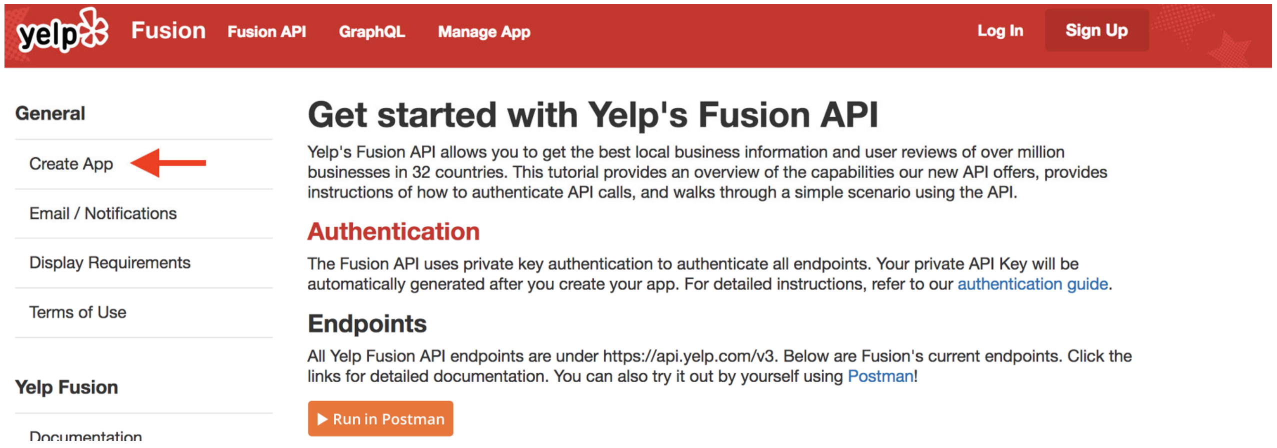
Name	Type	Description
total	int	Total number of business Yelp finds based on the search criteria. Sometimes, the value may exceed 1000. In such case, you still can only get up to 1000 businesses using multiple queries and combinations of the "limit" and "offset" parameters.
businesses	object[]	List of business Yelp finds based on the search criteria.
businesses[x].categories	object[]	List of category title and alias pairs associated with this business.
businesses[x].categories[x].alias	string	Alias of a category, when searching for business in certain categories, use alias rather than the title.
businesses[x].categories[x].title	string	Title of a category for display purpose.
businesses[x].coordinates	object	Coordinates of this business.
businesses[x].coordinates.latitude	decimal	Latitude of this business.
businesses[x].coordinates.longitude	decimal	Longitude of this business.
businesses[x].display_phone	string	Phone number of the business formatted nicely to be displayed to users. The format is the standard phone number format for the business's country.
businesses[x].distance	decimal	Distance in meters from the search location. This returns meters regardless of the locale.
businesses[x].id	string	Unique Yelp ID of this business. Example: '4kMBvIEWPxWkWKFN__8SxQ'
businesses[x].alias	string	Unique Yelp alias of this business. Can contain unicode characters. Example: 'yelp-san-francisco'. Also see What's the difference between the Yelp business ID and business alias?
businesses[x].image_url	string	URL of photo for this business.
businesses[x].is_closed	bool	Whether business has been (permanently) closed
businesses[x].location	object	Location of this business, including address, city, state, zip code and country.



businesses[x].location.address1	string	Street address of this business.
businesses[x].location.address2	string	Street address of this business, continued.
businesses[x].location.address3	string	Street address of this business, continued.
businesses[x].location.city	string	City of this business.
businesses[x].location.country	string	ISO 3166-1 alpha-2 country code of this business.
businesses[x].location.display_address	string[]	Array of strings that if organized vertically give an address that is in the standard address format for the business's country.
businesses[x].location.state	string	ISO 3166-2 (with a few exceptions) state code of this business.
businesses[x].location.zip_code	string	Zip code of this business.
businesses[x].name	string	Name of this business.
businesses[x].phone	string	Phone number of the business.
businesses[x].price	string	Price level of the business. Value is one of $, $$, $$$ and $$$$.
businesses[x].rating	decimal	Rating for this business (value ranges from 1, 1.5, ... 4.5, 5).
businesses[x].review_count	int	Number of reviews for this business.
businesses[x].url	string	URL for business page on Yelp.
businesses[x].transactions	string[]	List of Yelp transactions that the business is registered for. Current supported values are **pickup**, **delivery** and **restaurant_reservation**.
region	dict	Suggested area in a map to display results in.
region.center	dict	Center position of map area.
region.center.latitude	decimal	Latitude position of map bounds center.
region.center.longitude	decimal	Longitude position of map bounds center.

HOW TO ACCESS YELP API

https://www.yelp.com/developers/documentation/v3/authentication



Get started with Yelp's Fusion API

Yelp's Fusion API allows you to get the best local business information and user reviews of over million businesses in 32 countries. This tutorial provides an overview of the capabilities our new API offers, provides instructions of how to authenticate API calls, and walks through a simple scenario using the API.

Authentication

The Fusion API uses private key authentication to authenticate all endpoints. Your private API Key will be automatically generated after you create your app. For detailed instructions, refer to our authentication guide.

Endpoints

All Yelp Fusion API endpoints are under https://api.yelp.com/v3. Below are Fusion's current endpoints. Click the links for detailed documentation. You can also try it out by yourself using Postman!

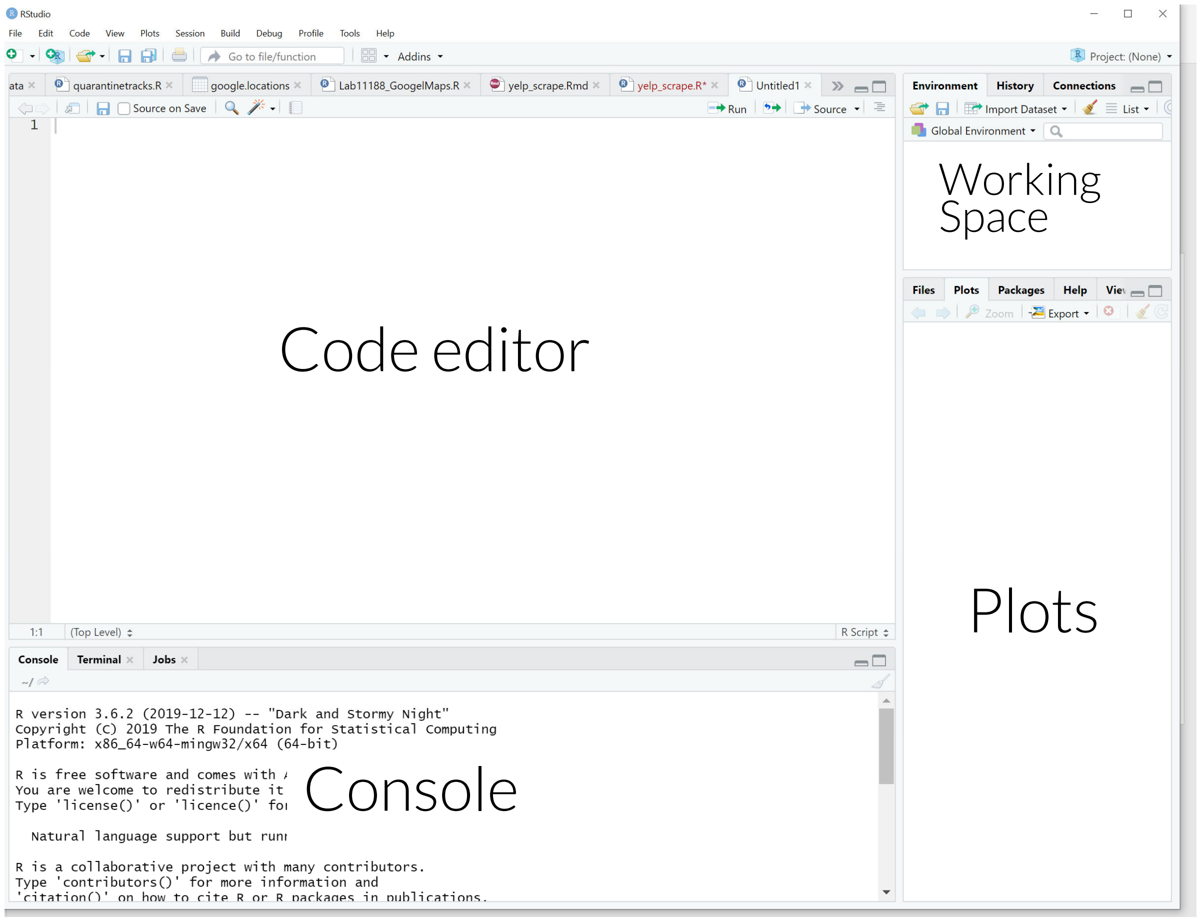
MORE INFORMATION ON YELP API

Documentation on Yelp API

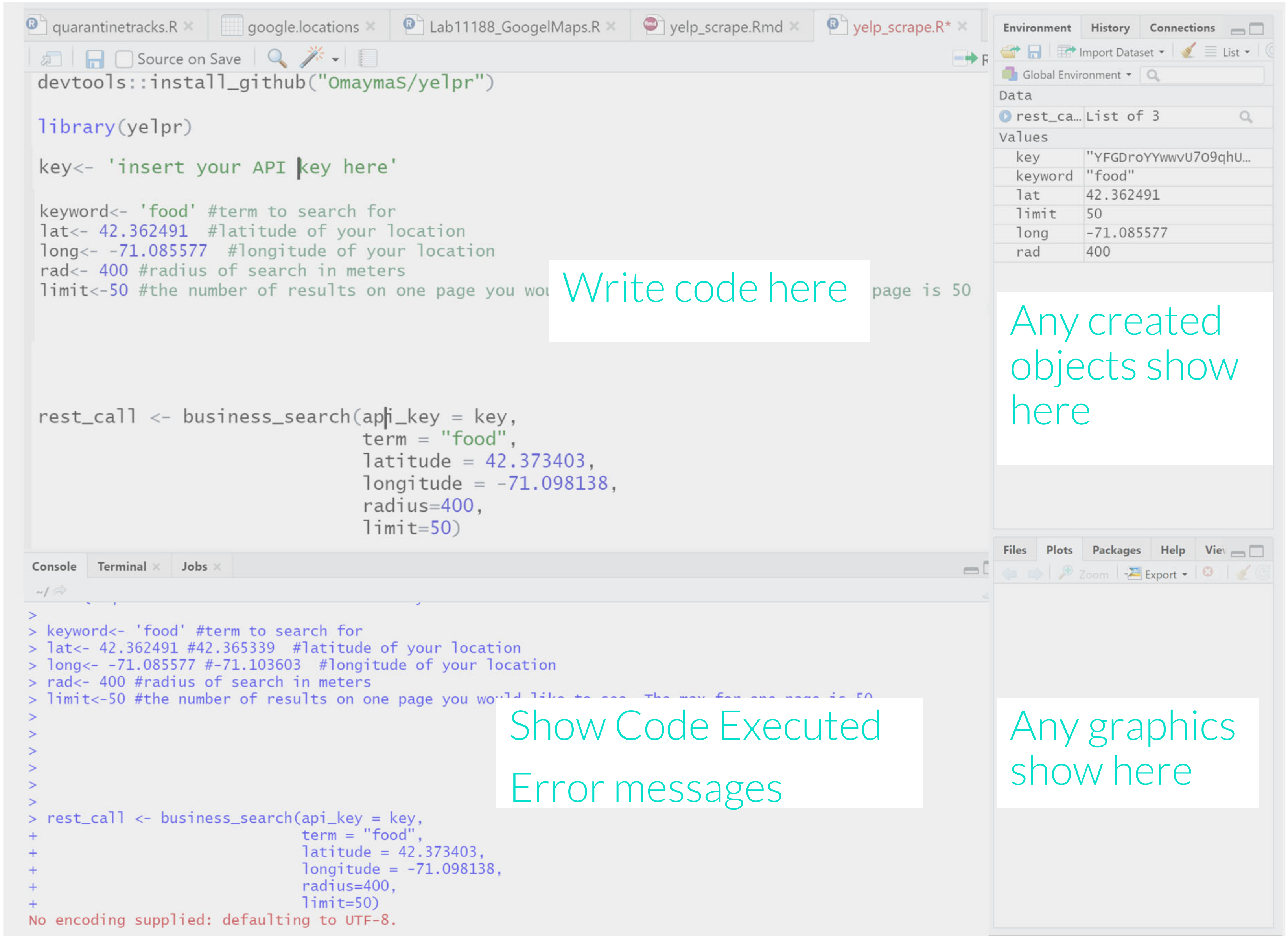
https://www.yelp.com/developers/documentation/v3/business_search

For detailed instructions on how to access the API refer to Lab 7 Part 1

R STUDIO



```
R version 3.6.2 (2019-12-12) -- "Dark and Stormy Night"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with /
You are welcome to redistribute it
Type 'license()' or 'licence()' for

  Natural language support but runn

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```



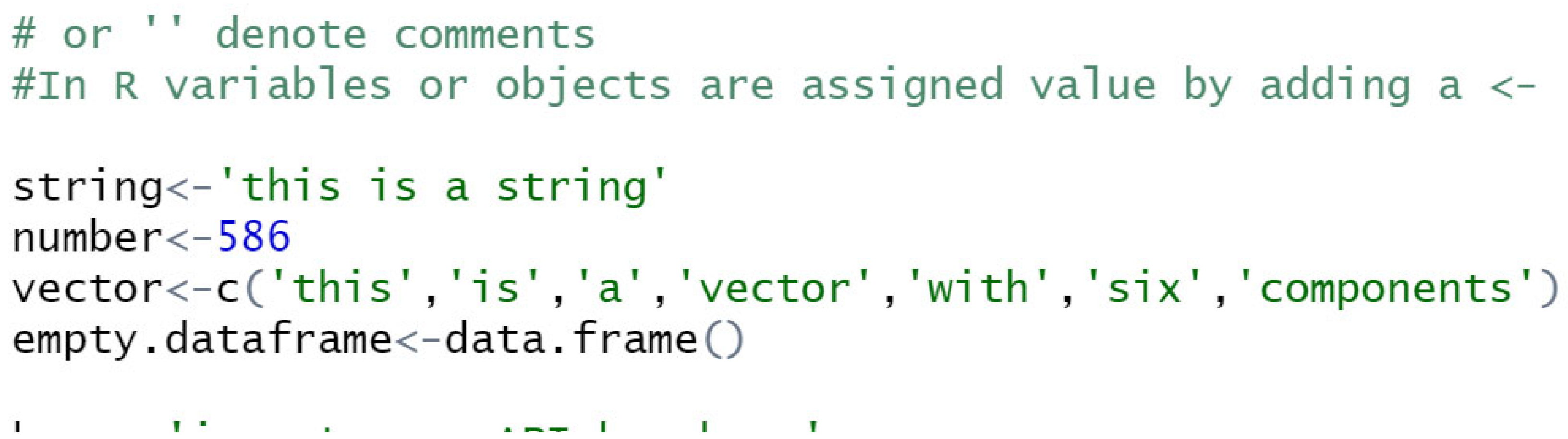
```r
devtools::install_github("OmaymaS/yelpr")

library(yelpr)

key<- 'insert your API key here'

keyword<- 'food' #term to search for
lat<- 42.362491  #latitude of your location
long<- -71.085577  #longitude of your location
rad<- 400 #radius of search in meters
limit<-50 #the number of results on one page you wou                         page is 50




rest_call <- business_search(api_key = key,
                             term = "food",
                             latitude = 42.373403,
                             longitude = -71.098138,
                             radius=400,
                             limit=50)
```

```
Data
rest_ca… List of 3
Values
  key       "YFGDroYYwwvU7O9qhU…
  keyword   "food"
  lat       42.362491
  limit     50
  long      -71.085577
  rad       400
```

```
~/
>
> keyword<- 'food' #term to search for
> lat<- 42.362491 #42.365339  #latitude of your location
> long<- -71.085577 #-71.103603  #longitude of your location
> rad<- 400 #radius of search in meters
> limit<-50 #the number of results on one page you wo
>
>
>
>
>
>
> rest_call <- business_search(api_key = key,
+                              term = "food",
+                              latitude = 42.373403,
+                              longitude = -71.098138,
+                              radius=400,
+                              limit=50)
No encoding supplied: defaulting to UTF-8.
```

WHAT’S HAPPENING IN THE CODE?

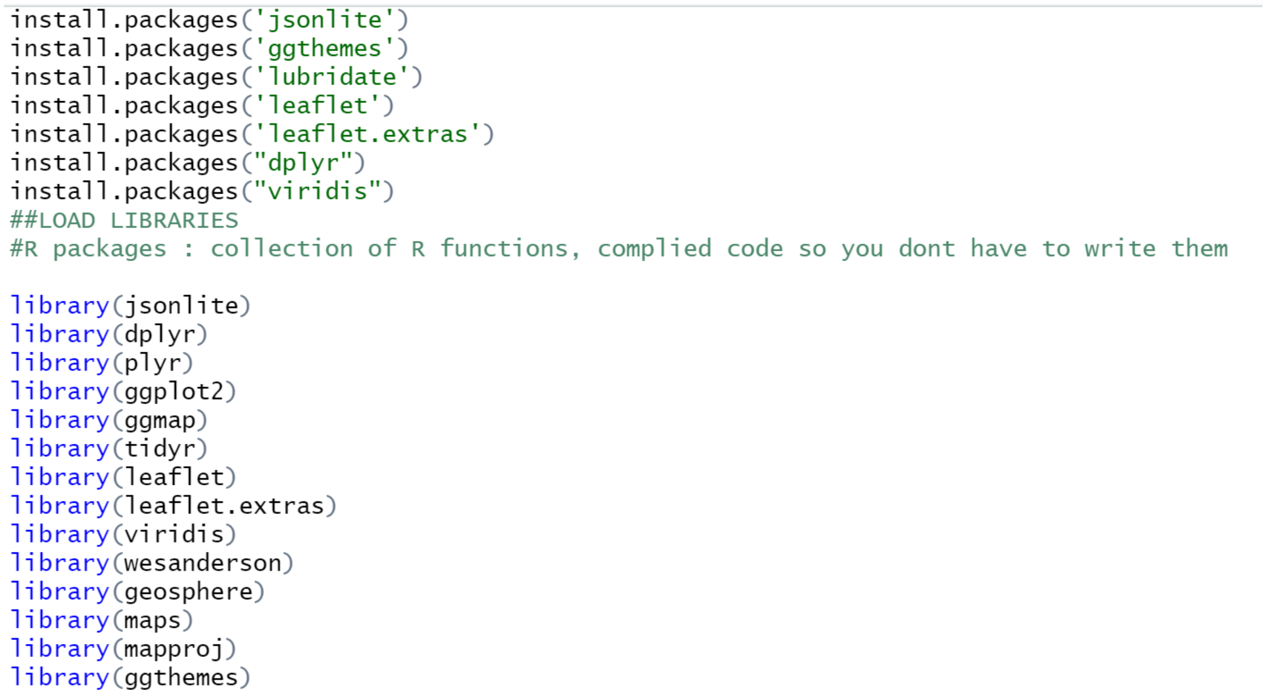
R Syntax



```r
# or '' denote comments
#In R variables or objects are assigned value by adding a <-

string<-'this is a string'
number<-586
vector<-c('this','is','a','vector','with','six','components')
empty.dataframe<-data.frame()
```

WHAT'S HAPPENING IN THE CODE?

Libraries and packages



```
install.packages('jsonlite')
install.packages('ggthemes')
install.packages('lubridate')
install.packages('leaflet')
install.packages('leaflet.extras')
install.packages("dplyr")
install.packages("viridis")
##LOAD LIBRARIES
#R packages : collection of R functions, complied code so you dont have to write them

library(jsonlite)
library(dplyr)
library(plyr)
library(ggplot2)
library(ggmap)
library(tidyr)
library(leaflet)
library(leaflet.extras)
library(viridis)
library(wesanderson)
library(geosphere)
library(maps)
library(mapproj)
library(ggthemes)
```

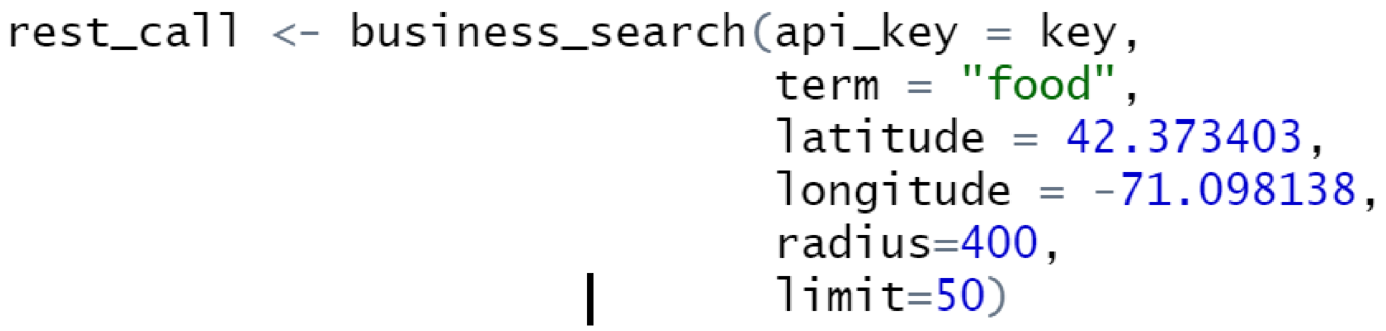
WHAT'S HAPPENING IN THE CODE?

Set Parameters for the search

```
key<- 'insert your API key here'


keyword<- 'food' #term to search for
lat<- 42.362491  #latitude of your location
long<- -71.085577  #longitude of your location
rad<- 400 #radius of search in meters
limit<-50 #the number of results on one page you would like to see. The max for one page is 50
```

WHAT'S HAPPENING IN THE CODE?

Make a call to the API using a function called 'business_search' in library yelpR

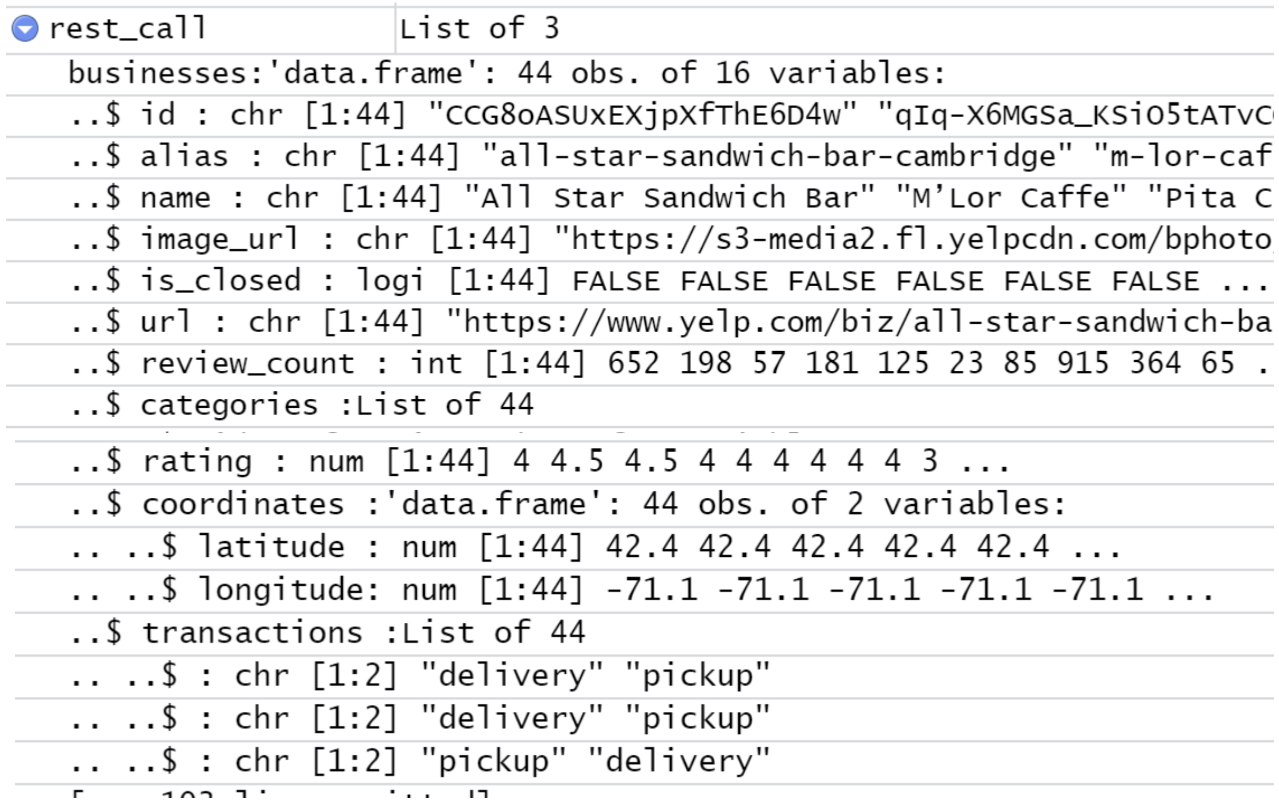


```r
rest_call <- business_search(api_key = key,
                             term = "food",
                             latitude = 42.373403,
                             longitude = -71.098138,
                             radius=400,
                             limit=50)
```

WHAT'S HAPPENING IN THE CODE?

API call returns a list with information of restaurants.

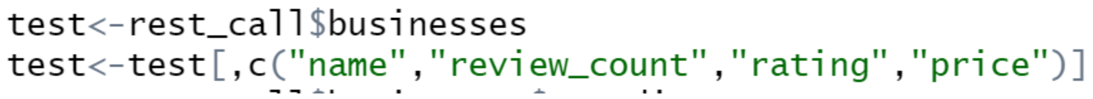
(remember : *1 call only returns 50 results-then we have to make another call to 'get the next page of results')*



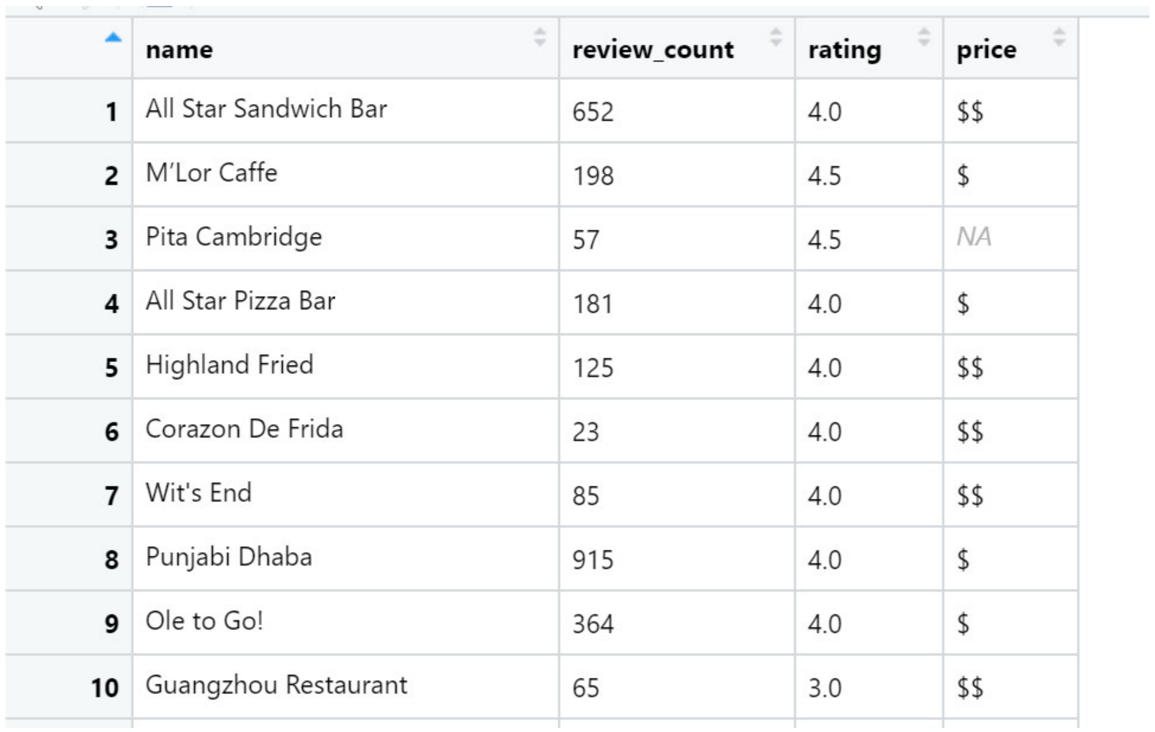
```
rest_call                    List of 3
  businesses:'data.frame': 44 obs. of 16 variables:
  ..$ id : chr [1:44] "CCG8oASUxEXjpXfThE6D4w" "qIq-X6MGSa_KSiO5tATvC
  ..$ alias : chr [1:44] "all-star-sandwich-bar-cambridge" "m-lor-caf
  ..$ name : chr [1:44] "All Star Sandwich Bar" "M'Lor Caffe" "Pita C
  ..$ image_url : chr [1:44] "https://s3-media2.fl.yelpcdn.com/bphoto
  ..$ is_closed : logi [1:44] FALSE FALSE FALSE FALSE FALSE FALSE ...
  ..$ url : chr [1:44] "https://www.yelp.com/biz/all-star-sandwich-ba
  ..$ review_count : int [1:44] 652 198 57 181 125 23 85 915 364 65 .
  ..$ categories :List of 44
  ..$ rating : num [1:44] 4 4.5 4.5 4 4 4 4 4 4 3 ...
  ..$ coordinates :'data.frame': 44 obs. of 2 variables:
  .. ..$ latitude : num [1:44] 42.4 42.4 42.4 42.4 42.4 ...
  .. ..$ longitude: num [1:44] -71.1 -71.1 -71.1 -71.1 -71.1 ...
  ..$ transactions :List of 44
  .. ..$ : chr [1:2] "delivery" "pickup"
  .. ..$ : chr [1:2] "delivery" "pickup"
  .. ..$ : chr [1:2] "pickup" "delivery"
```

WHAT’S HAPPENING IN THE CODE?

Extract certain elements on list we want and turn it into a dataframe



```
test<-rest_call$businesses
test<-test[,c("name","review_count","rating","price")]
```



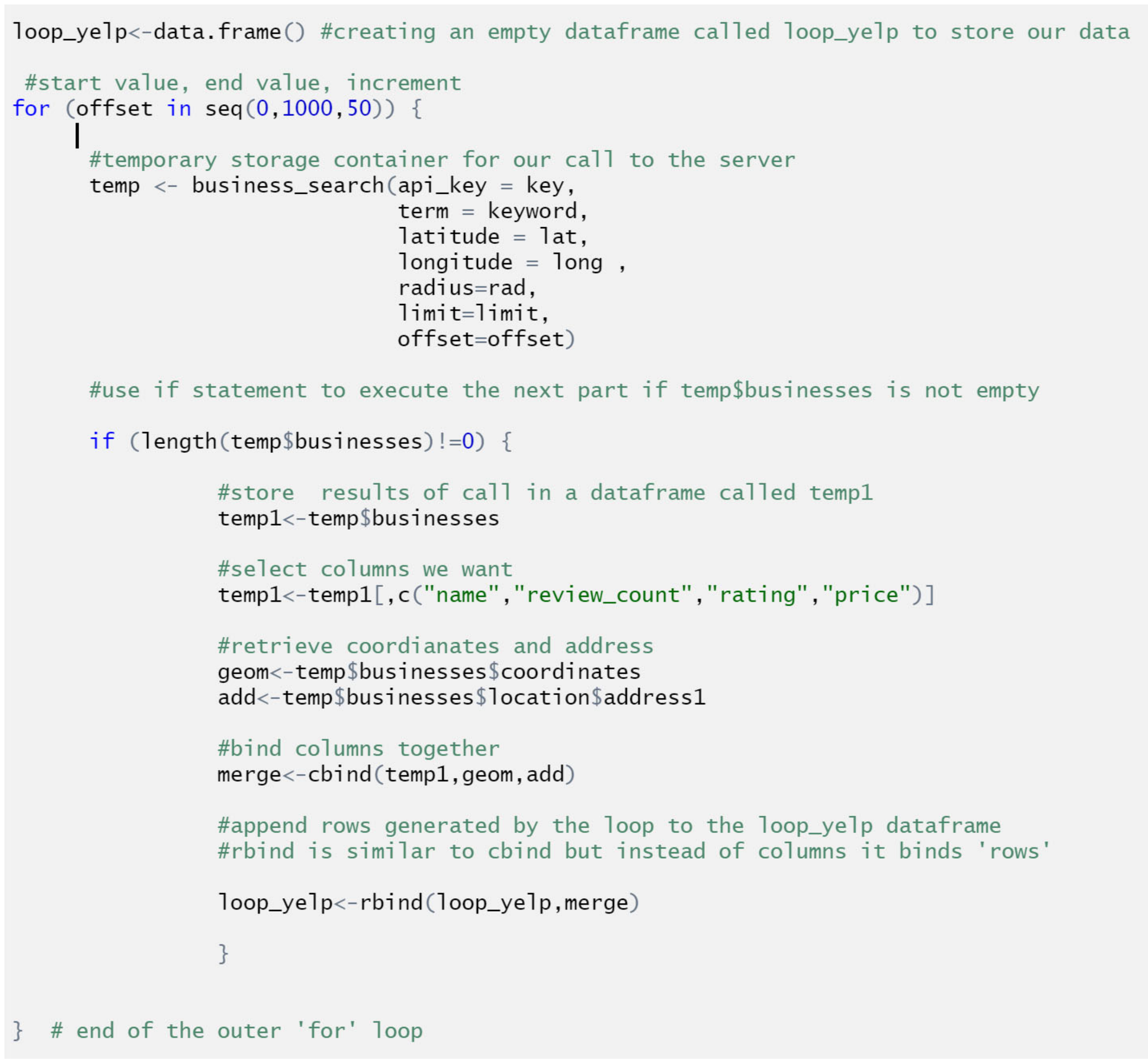
	name	review_count	rating	price
1	All Star Sandwich Bar	652	4.0	$$
2	M’Lor Caffe	198	4.5	$
3	Pita Cambridge	57	4.5	NA
4	All Star Pizza Bar	181	4.0	$
5	Highland Fried	125	4.0	$$
6	Corazon De Frida	23	4.0	$$
7	Wit's End	85	4.0	$$
8	Punjabi Dhaba	915	4.0	$
9	Ole to Go!	364	4.0	$
10	Guangzhou Restaurant	65	3.0	$$

Make a loop that repeats this process

Creates a variable called offset to get ‘next page’
Which starts at 0
Then 50 [to get results 51-100]
Then 100 [to get results 101-151]
And so on till empty results

Create a storage container that stores each new call results

Then appends the results to final dataframe



```r
loop_yelp<-data.frame() #creating an empty dataframe called loop_yelp to store our data

 #start value, end value, increment
for (offset in seq(0,1000,50)) {

      #temporary storage container for our call to the server
      temp <- business_search(api_key = key,
                              term = keyword,
                              latitude = lat,
                              longitude = long ,
                              radius=rad,
                              limit=limit,
                              offset=offset)

      #use if statement to execute the next part if temp$businesses is not empty

      if (length(temp$businesses)!=0) {

                #store  results of call in a dataframe called temp1
                temp1<-temp$businesses

                #select columns we want
                temp1<-temp1[,c("name","review_count","rating","price")]

                #retrieve coordianates and address
                geom<-temp$businesses$coordinates
                add<-temp$businesses$location$address1

                #bind columns together
                merge<-cbind(temp1,geom,add)

                #append rows generated by the loop to the loop_yelp dataframe
                #rbind is similar to cbind but instead of columns it binds 'rows'

                loop_yelp<-rbind(loop_yelp,merge)

                }


}  # end of the outer 'for' loop
```

DISADVANTAGES OF DIGITAL TRACE DATA

No control over what and is not available or understanding of how it is stored

- e.g. is an establishment showing up under the search term of ‘food’ vs ‘restaurants’

Validity of the inferences

- can only observe behavior, not understand intentionality
- behavior being observed on social media is not ‘natural’ or non-reactive’

Conflict with current standards of informed consent and privacy