**1. What is your package trying to achieve? Who is it aimed at?**

This R package will serve as a restaurant finder. Given any location and some restaurant preferences, this package will return the restaurants closest to the search preferences. We will also plot the restaurant locations on a map. Some possible search options would be around the cuisine, whether a restaurant offers deals, hours of operation, etc. This app is aimed at developers working in R, who may want to integrate local business information in their app/website.

**3. Briefly describe the background if that's not obvious. What field or area of interest is it related to? (Just briefly, in a paragraph or two, maximum.) References, including web references, are welcome where available.**

We wanted to create a social/local experience package in R. From amongst the data sets available, we found 2 data sets useful to use coming from yelp and factual that provide local business information. The scenario chosen would be to find a business close to a user based on some user preferences on the business. To narrow down our scenario, we choose restaurants that has more data coming from yelp. We plan to integrate the yelp data and data avaialble from factual to provide an R package that helps a user quickly locate a restaurant.

Upon investigation, we found that there are no R package that implements restaurant finder functionality. By combining data from 2 data sources, we feel that we will provide enriched information about local restaurants.

Each local business type from our two sources has different data-schema, and different user scenarios. Due to time constraints, we are focusing primarily on restaurants for this quarter, but later, this package can be extended to cover other local businesses too.

**4. Provide a few use cases; that is, in a sentence or so say how you see someone using the package.**

1. Search top 10 (or any other user-configurable number) restaurants in a geographical area by

a. Cuisine type

b. Name (Individual restaurant name or chains like McDonald's or Subway)

d. Has Deals

e. Distance from a certain address (Walk-score)

f. All restaurants that offer delivery of food given a location and distance from user

g. Price range

h. Dietary preferences: vegan, vegetarian etc.

i. Other preferences: Alcohol availability, attire, parking options etc.

For each of the scenarios above, provide restaurant name, cusine type, deal info, address, open hrs and other related information, review information.

These functions will return a table, which the user can further analyze in an R session, export to an external file or display in a browser.

2. Plot businesses returned in functions in (1) on a map.

3. Analyze the restaurant data returned by functions in 1 from both sources, aggregate the two streams, and sort the restaurants through a rule-based recommendation scheme based on distance, review count and average rating.

This is the tentative list of functions that we plan to implement:

1. Search functions:

a) getRestaurantByCusine(cusineType, areaName, numResults)

b) getRestaurantByName(name, areaName, numResults) (Make it list of restaurants)

c) getRestauratsOpen(DateTime, areaName, cusine, numResults) (Low)

d) getRestauratsWithDeals(areaName, cusine, numResults) (Yelp)

e) getNearestRestaurant(address, cuisine, numResults) (Make it specific to attractions)

f) getRestaurantsWithDelivery(address, maxdistance, cusine, numResults) (Low)

g) getRestaurantByPrice(area, cuisine, numResults)

h) GetRestaurantByDietType(dietType, areaName, cusine, numResults)

i) Other preferences:

GetAlcoholFriendlyRestaurants(areaName, cuisine, numResults)

GetRestaurantByAttire(attireType, areaName, cuisine, numResults)

GetRestaurantByParking(parkingType, areaName, cuisine, numResults)

numResults will have default value of 10 unless changed by the user.

cusineType will be All unless changed by the user.

2. Map-plotting funcion:

PlotBusinesses(NameLatLongDF)

NameLatLongDF will be a dataframe containing the columns name, latitude and longitude of the information (returned in use cases 1) that needs to be plotted on the map

3. Sorting function: SortResults(restaurantResultDF)

restaurantResultDF will have the restaurant ids, ratingnumber and number of reviews, distance from user if available```

* Factual
  + Get company level information (employees)
* Top X in each state (cuisine, chains like Starbucks etc.) Make it compare functions() user provides a list
* Given a city map and a cuisine type, find business opportunity by marking biggest unrepresented areas for that cuisine
* On the map, click on an entity to get more details (Look for interactive map)
* Plot heat map on certain parameters (Dressing: formal, Vegetarian friendly)

Limit things to the US