Introduction

Download the file RestaurantData.zip file containing the data for this project assignment from the Moodle. Unzip the file in a directory that will serve as your working directory. When you start up R, make sure to change your working directory to the directory where you unzipped the data.

The data for this project came from Kaggle where somebody had used the Zomato API to scrape the data. Zomato API Analysis is useful for foodies who want to taste the best cuisines of every part of the world which lies in their budget.

Data Fetching: Data had been originally collected from the Zomato API in the form of .json files (raw data). The collected data has been stored in the Comma Separated Value file zomato.csv.

Each restaurant in the dataset is uniquely identified by its Restaurant Id. Every Restaurant contains the following variables:

- Restaurant Id: Unique id of every restaurant across various cities of the world
- Restaurant Name: Name of the restaurant
- Country Code: Country in which restaurant is located
- City: City in which restaurant is located
- Address: Address of the restaurant
- Locality: Location in the city
- Locality Verbose: Detailed description of the locality
- Longitude: Longitude coordinate of the restaurant's location
- Latitude: Latitude coordinate of the restaurant's location
- Cuisines: Cuisines offered by the restaurant
- Average Cost for two: Cost for two people in different currencies 👫
- Currency: Currency of the country
- Has Table booking yes/no
- Has Online delivery: yes/ no
- Is delivering yes/ no
- Switch to order menu: yes/no
- Price range: range of price of food
- Aggregate Rating: Average rating out of 5
- Rating color: depending upon the average rating color
- Rating text: text on the basis of rating of rating
- Votes: Number of ratings casted by people

Part 1 Plot the user rankings for the restaurants

- Read the Zomato data into R via the read_csv function and look at the first few rows
- There are many columns in this dataset. You can see how many by typing `ncol(zomato)`(you can see the number of rows with the nrow function).
- In addition, you can see the names of each column by typing `names(zomato)` (the names are also listed on the previous page.)

To make a simple histogram of the ratings (column 18 in the zomato dataset), run

```
> ## You may get Error in plot.new() : figure margins too large; make sure your
plotting panel in RStudio is large enough
> hist(zomato$`Aggregate rating`)
```

Part 2 Finding the 3 best restaurants in a country by price range

Write a function called ***best*** that takes two arguments: a 3-character country code of a country and a price range (numbers 1,2,3,4). The function should read the *zomato.csv* file and return a tibble with the names of the restaurant and the city that have the **3 best (i.e. highest) ranking for a price range in that country**. There are 4 price range levels.

- Countries that do not have data on a particular price range should be excluded from theset of restaurants when deciding the rankings.

- Handling ties. If there is a tie for the best restaurant for a given price range, then therestaurant names should be sorted in alphabetical order.

The function should use the following template.

```
best <- function(country, pricerange) {
        ## Read zomato data
        ## Check that country and price range are valid
        ## Return restaurant name in that state with highest ranking
}
```

- The function should check the validity of its arguments. If an invalid country value is passed to best, the function should throw an error via the stop function with the exact message **"invalid country"**.
- If an invalid price range value is passed to best, the function should throw an error via the stop function with the exact message **"invalid price range"**.

Here is some sample output from the function (WARNING: we will run different tests and the results shown here might not actually be correct)

```
> source("best.R")
> best("USA", 1)
# A tibble: 3 x 2
  `Restaurant Name`       City
  <chr>                   <chr>
1 Oakwood Cafe            Dalton
2 Rae's Coastal Cafe      Augusta
3 Shorts Burger and Shine Cedar Rapids/Iowa City
> best("IND", 3)
# A tibble: 3 x 2
  `Restaurant Name`         City
  <chr>                     <chr>
1 AB's - Absolute Barbecues Chennai
2 Sagar Gaire Fast Food     Bhopal
3 Sheroes Hangout           Agra
> best("TGH", 4)
```

```
Error in best("TGH", 4): invalid country
> best("BRA", 5)
Error in best("BRA", 5) : invalid price range>
```

Save your code for this function to a file named best.R and submit this file for Part1 & 2.

Part 3 Ranking restaurants by price range in a country

Write a function called ***rankrestaurant*** that takes three arguments: the 3-character countrycode of a country (*country*), a price range (*pricerange*), and the ranking of a restaurant in that country for that price range (*num*). The function should read the *zomato.csv* file and return a tibble with the name of the restaurant and the city that has the ranking specified by the *num* argument.

For example, the call

rankrestaurant("IND", 1, 5)

would return a tibble containing the name of the restaurant and city with the 5th best user ranking.

- The *num* argument can take values "best", "worst", or an integer indicating the ranking (higher numbers are better). If the number given by *num* is larger than the number of restaurants in that country, then the function should return NA.
- Restaurants that do not have data on a price range should be excluded from the set of restaurants when deciding the rankings.

Handling ties. It may occur that multiple restaurants have the same user rankings. In those cases, ties should be broken by using the restaurant name. For example, in India ("IND"), the restaurants in price range 4 with aggregate rating are shown here.

```
# A tibble: 10 x 2
   `Restaurant Name`           `Aggregate rating`
   <chr>                                    <dbl>
 1 AB's - Absolute Barbecues                  4.9
 2 AB's - Absolute Barbecues                  4.9
 3 AB's - Absolute Barbecues                  4.9
 4 Barbeque Nation                            4.9
 5 CakeBee                                    4.9
 6 Caterspoint                                4.9
 7 Sagar Gaire Fast Food                      4.9
 8 Sheroes Hangout                            4.9
 9 The Great Indian Pub                       4.9
10 Zolocrust - Hotel Clarks Amer              4.9
```

Note that all have the same aggregate rankings. However, they are listed alphabetical here.
You can use **the order function** to sort multiple vectors (because of multiple groups of ranking values) in this manner (i.e. where one vector is used to break ties in another vector). You can achieve the same by using functions from **dplyr** as well.

The function should use the following template.

```
rankrestaurant <- function(country, pricerange, num = "best") {
        ## Read zomato data
        ## Check that country, pricerange and num are valid
        ## Return restaurant with num ranking
}
```

The function should check the validity of its arguments.

- If an invalid country value is passed to best, the function should throw an error via the stop function with the exact message “invalid country”.
- If an invalid price range value is passed to best, the function should throw an error via the stop function with the exact message “invalid price range”.

Here is some sample output from the function.

```
> source("rankrestaurant.R")
> rankrestaurant(“BRA”,2, "best")
# A tibble: 1 x 2
  `Restaurant Name` City
  <chr>             <chr>
1 Leme Light        Rio de Janeiro
> rankrestaurant("USA", 2, "worst")
# A tibble: 1 x 2
  `Restaurant Name` City
  <chr>             <chr>
1 Nosh Mahal        Pocatello
> rankrestaurant("IND", 4, 1400)
# A tibble: 1 x 2
  `Restaurant Name` City
  <chr>             <chr>
1 NA                NA
> rankrestaurant("IND", 4, 30)
# A tibble: 1 x 2
  `Restaurant Name` City
  <chr>             <chr>
1 That Place        Vadodara
```

Save your code for this function to a file named rankrestaurant.R and submit the file.

Part 4 Ranking restaurants in all countries

Write a function called ***rankall*** that takes two arguments: a price range (*pricerange*) and arestaurant ranking (*num*). The function reads the *zomato.csv* file and returns a tibble containing the restaurant in each country that has the ranking specified in num. For example the function call *rankall(1, "best")* would return a tibble containing the names of the restaurants that are the best in their respective price range. The function should return a value for every country (some may be NA).

The first column in the data frame should be named *restaurant,* which contains the restaurant name, and the second column should be named *country*, which contains the 3-character country code.

Restaurants that do not have data on a particular price range should be excluded from the set of restaurants when deciding the rankings.

Handling ties. The *rankall* function should handle ties in the same way that the *rankrestaurant* function handles ties.

The function should use the following template.

```
rankall <- function(pricernage, num = "best") {
        ## Read zomato data
        ## Check that pricerange and num are valid
        ## For each country, find the restaurant of the given rank
        ## Return a tibble with the restaurant names and the country code
}
```

NOTE: For the purpose of this part of the assignment (and for efficiency), your function should NOT call the rankrestaurant function from the previous section.

The function should check the validity of its arguments. If an invalid price range value is passed to rankall, the function should throw an error via the stop function with the exact message “invalid price range”. The num variable can take values “best”, “worst”, or an integer indicating the ranking (smaller numbers are better). If the number given by num is larger than the number of restaurants in that state, then the function should return NA.

Here is some sample output from the function.

```
> source("rankall.R")
> head(rankall(4, 20), 10)
      restaurant country
7  TGI Friday's     ARE
4          <NA>     AUS
2    Coco Bambu     BRA
6          <NA>     CAN
10         <NA>     GBR
8       Zaffran     IND
13         <NA>     LKA
9          <NA>     NZL
1          <NA>     PHL
11         <NA>     QAT

> tail(rankall(3, "worst"), 3)
restaurant country
14 Leman K\xed_lt\xed_r     TUR
3       The Lady & Sons     USA
12             Parrot's     ZAF


> tail(rankall(1), 5)
                     restaurant country
    1 Caf\xed\xa9 Daniel Briand     BRA
    6              Ju Ju's Cafe     GBR
    4 Grandson of Tunday Kababi     IND
    5                     Miann     NZL
    2   Ingleside Village Pizza     USA
```

Save your code for this function to a file named *rankall.R* and submit.

Submission:

Please submit 1 zip file that contains all 3 *.R code files.