```
import json

# Open the JSON file in read mode
with open("data.json", "r") as file:
    # Load the JSON data into a Python dictionary
    data = json.load(file)
```

Geeks for Geeks

Problem 1: Swiggy Restaurants Data Analysis

Submitted by Abhilash V A - 9880507105 - aabhilash070@gmail.com

```
data.keys()
```

```
dict_keys(['Abohar', 'Adilabad', 'Adityapur', 'Adoni', 'Agartala', 'Agra', 'Ahmedabad', 'Ahmednagar', 'Aizawl', 'Ajmer', 'Akola', 'Alappuzha', 'Aligarh', 'Alipurduar', 'Allahabad',
'Almora', 'Alwar', 'Amalapuram', 'Ambala', 'Ambikapur', 'Ambur', 'Amravati', 'Amreli', 'Amritsar', 'Anand', 'Anantapur', 'Angul', 'Ankleshwar', 'Arakkonam', 'Arambagh', 'Arrah',
'Aruppukottai', 'Asansol', 'Aurangabad', 'Aurangabad_bihar', 'Azamgarh', 'Baddi', 'Bagalkot', 'Bagdogra', 'Bagula', 'Bahadurgarh', 'Bahraich', 'Balaghat', 'Balangir', 'Balasore',
'Ballari', 'Balrampur', 'Balurghat', 'Banda', 'Bangalore', 'Bankabihar', 'Bankura', 'Bantwal', 'Bapatlachirala', 'Baramati', 'Baran', 'Baraut', 'Bardhaman', 'Bardoli', 'Bareilly',
'Baripada', 'Barmer', 'Barnala', 'Barshi', 'Barwani', 'Basirhat', 'Basti', 'Batala', 'Bathinda', 'Beawar', 'Beed', 'Begusarai', 'Bela-pratapgarh', 'Belgaum', 'Berhampore',
'Berhampur', 'Bettiah', 'Betul', 'Bhadohi', 'Bhadrachalam', 'Bhadrak', 'Bhadravati', 'Bhagalpur', 'Bhandara', 'Bharabanki', 'Bharatpur', 'Bharuch', 'Bhatkal', 'Bhavnagar',
'Bhawanipatna', 'Bhilai', 'Bhilwara', 'Bhimavaram', 'Bhind', 'Bhiwadi', 'Bhiwani', 'Bhopal', 'Bhubaneswar', 'Bhuj', 'Bhusawal', 'Bidar', 'Biharsharif', 'Bijapur', 'Bijnor', 'Bikaner',
'Bilaspur', 'Bilaspur-hp', 'Bilimora', 'Biswanath-chariali', 'Bodhan-rural', 'Bodinayakanur', 'Boisar', 'Bokaro', 'Bolpur', 'Bongaigaon', 'Bongaon', 'Budaun', 'Budhwal',
'Bulandshahr', 'Buldana', 'Bundi', 'Burhanpur', 'Buxar', 'Central-goa', 'Chakdaha', 'Chalakkudy', 'Chalisgaon', 'Chandausi', 'Chandigarh', 'Chandrapur', 'Changanassery', 'Chengannur',
'Chennai', 'Cherthala', 'Chhapra', 'Chhatarpur', 'Chhindwara', 'Chidambaram', 'Chikhli', 'Chikkaballapur', 'Chikmagalur', 'Chiplun', 'Chitradurga', 'Chittoor', 'Chittorgarh',
'Chopda', 'Churu', 'Coimbatore', 'Cooch-behar', 'Cuddalore', 'Cuttack', 'Dahanu', 'Dahod', 'Daltonganj', 'Daman', 'Darbhanga', 'Darjeeling', 'Daund', 'Dausa', 'Davanagere',
'Dehradun', 'Dehri', 'Delhi', 'Deoghar', 'Dewas', 'Dhanbad', 'Dhar', 'Dharamshala', 'Dharapuram', 'Dharmapuri', 'Dharwad', 'Dhoraji', 'Dhule', 'Dibrugarh', 'Digboi', 'Dimapur',
'Dindigul', 'Diu', 'Doddaballapura', 'Duliajan', 'Dumka', 'Durgapur', 'Eluru', 'Erode', 'Etah', 'Etawah', 'Faizabad', 'Faridabad', 'Faridkot', 'Farrukhabad', 'Fatehabad', 'Fatehgarh-
sahib', 'Fatehpur', 'Fazilka', 'Firozabad', 'Firozpur', 'Freelancer', 'Gadag-betigeri', 'Gadwal', 'Gandhidham', 'Gangapur-city', 'Gangarampur', 'Gangtok', 'Gauriganj', 'Gaya',
'Giridih', 'Godhra', 'Gokak', 'Golaghat', 'Gonda', 'Gondal', 'Gondia', 'Gopalganj', 'Gorakhpur', 'Gudivada', 'Guna', 'Guntakal', 'Guntur', 'Gurdaspur', 'Gurgaon', 'Guwahati',
'Gwalior', 'Habra', 'Hajipur', 'Haldia', 'Haldwani', 'Halol', 'Hampi', 'Hansi', 'Hanumangarh', 'Hapur', 'Hardoi', 'Haridwar', 'Hassan', 'Hathras', 'Himmatnagar', 'Hindaun',
'Hinganghat', 'Hisar', 'Hoshangabad', 'Hoshiarpur', 'Hospet', 'Hosur', 'Hubli', 'Hyderabad', 'Ichalkaranji', 'Idukki', 'Imphal', 'Indore', 'Irinjalakuda', 'Itanagar', 'Itarsi',
'Jabalpur', 'Jagdalpur', 'Jagraon', 'Jagtial', 'Jahanabad', 'Jaigaon', 'Jaipur', 'Jalandhar', 'Jalaun', 'Jalgaon', 'Jalpaiguri', 'Jammu', 'Jamnagar', 'Jamshedpur', 'Jaunpur',
'Jhalawar', 'Jhansi', 'Jhargram', 'Jhunjhunu', 'Jind', 'Jodhpur', 'Jorhat', 'Junagadh', 'Kadapa', 'Kadayanallur', 'Kadiri', 'Kaithal', 'Kakinada', 'Kalaburagi', 'Kalady', 'Kalna',
'Kamareddy', 'Kanchrapara', 'Kannauj', 'Kannur', 'Kanpur', 'Kanyakumari', 'Kapurthala', 'Karad', 'Karaikkudi', 'Karimnagar', 'Karnal', 'Karunagappaly', 'Karur', 'Karwar', 'Kasaragod',
'Kashipur', 'Katihar', 'Katni', 'Kavali', 'Kayamkulam', 'Kendrapada', 'Kendujhar', 'Khalilabad', 'Khamgaon', 'Khammam', 'Khandwa', 'Khanna', 'Kharagpur', 'Khopoli', 'Kishanganj',
'Kishangarh', 'Kochi', 'Kodaikanal', 'Kohima', 'Kolar', 'Kolhapur', 'Kolkata', 'Kollam', 'Kopargaon', 'Koppal', 'Korba', 'Kota', 'Kotdwar', 'Kothagudem', 'Kothamanagalam',
'Kottakkal', 'Kottarakkara', 'Kottayam', 'Kovilpatti', 'Kozhikode', 'Krishnagiri', 'Krishnanagar', 'Kumarakom', 'Kumbakonam', 'Kumta', 'Kundapura', 'Kunnamkullam', 'Kurnool',
'Kurukshetra', 'Lakhimpur', 'Lalitpur', 'Latur', 'Lonavala', 'Lonavla', 'Lucknow', 'Ludhiana', 'Machilipatnam', 'Madanapalle', 'Madhubani', 'Madikeri', 'Madurai', 'Mahabaleshwar',
'Mahbubnagar', 'Mahoba', 'Malappuram', 'Malda', 'Malegaon', 'Malout', 'Manali', 'Mancherial', 'Mandi-dabwali', 'Mandi-gobindgarh', 'Mandi-hp', 'Mandsaur', 'Mandya', 'Mangaluru',
'Manipal', 'Manjeri', 'Mannargudi', 'Mansa', 'Markapur', 'Mathura', 'Maunath-bhanjan', 'Mayiladuthurai', 'Medinipur', 'Meerut', 'Mehsana', 'Mettupalayam', 'Miryalaguda', 'Mirzapur',
'Modinagar', 'Moga', 'Moodbidri', 'Moradabad', 'Morbi', 'Morena', 'Motihari', 'Mount-abu', 'Mughalsarai', 'Mukerian', 'Muktsar', 'Mumbai', 'Munger', 'Murshidabad', 'Mussoorie',
'Muvattupuzha', 'Muzaffarnagar', 'Muzaffarpur', 'Mysore', 'Nabadwip', 'Nadiad', 'Nagaon', 'Nagapattinam', 'Nagda', 'Nagercoil', 'Nagpur', 'Nagur', 'Naharlagun', 'Nainital', 'Nalbari',
'Nalgonda', 'Namakkal', 'Nanded', 'Nandurbar', 'Nandyal', 'Nangal', 'Naraingarh', 'Narasaraopet', 'Narnaul', 'Narsinghpur', 'Narsipatnam', 'Nashik', 'Navsari', 'Neemuch', 'Nellore',
'Neyveli', 'Nipani', 'Nirmal', 'Nizamabad', 'Noida', 'Noida-1', 'North-goa', 'Ongole', 'Ooty', 'Orai', 'Osmanabad', 'Pala', 'Palakkad', 'Palakollu', 'Palampur', 'Palani', 'Palanpur',
'Pali', 'Palwal', 'Panchgani', 'Panipat', 'Paramakudi', 'Parbhani', 'Pathankot', 'Patiala', 'Patna', 'Pattukkottai', 'Payyannur', 'Perambalur', 'Perinthalmanna', 'Phagwara', 'Phusro',
'Pilani', 'Pilibhit', 'Pinjore-city', 'Pondicherry', 'Porbandar', 'Port-blair', 'Pratapgarh', 'Proddatur', 'Pudukkottai', 'Pune', 'Puri', 'Purnea', 'Purulia', 'Pusad', 'Puttur',
'Puttur-karnataka', 'Rae-bareli', 'Raghunathpur', 'Raichur', 'Raiganj', 'Raigarh', 'Raipur', 'Rajahmundry', 'Rajapalayam', 'Rajgarh', 'Rajkot', 'Rajnandgaon', 'Rajsamand',
'Ramagundam', 'Ramanagara', 'Ramanathapuram', 'Ramgarh', 'Rampur', 'Ranaghat', 'Ranaghat-wb', 'Ranchi', 'Rangpo', 'Ranibennur', 'Raniganj', 'Ratlam', 'Ratnagiri', 'Ravulapalem',
```

```
'Rayachoty', 'Rewa', 'Rewari', 'Rishikesh', 'Rohtak', 'Roorkee', 'Ropar', 'Rourkela', 'Rudrapur', 'Sagar', 'Saharanpur', 'Saharsa', 'Salem', 'Samastipur', 'Sambalpur', 'Sangamner',
'Sangli', 'Sangrur', 'Santipur', 'Sasaram', 'Satara', 'Satna', 'Sawai-madhopur', 'Sawantwadi', 'Sehore', 'Shahjahanpur', 'Shamli', 'Shikohabad', 'Shillong', 'Shimla', 'Shirdi-city',
'Shivamogga', 'Shivpuri', 'Shravasti', 'Shrirampur', 'Siddipet', 'Sikar', 'Silchar', 'Siliguri', 'Silvassa', 'Sindhanur', 'Singrauli', 'Sirkali', 'Sirsa', 'Sirsi', 'Sitamarhi',
'Sitapur', 'Sivakasi', 'Sivasagar', 'Siwan', 'Solan', 'Solapur', 'Sonipat', 'South-goa', 'Sri-ganganagar', 'Srikakulam', 'Srivilliputhur', 'Sultanpur', 'Surat', 'Suratgarh',
'Surendranagar-dudhrej', 'Suri', 'Suryapet', 'Tadepalligudem', 'Tadpatri', 'Tanuku', 'Tarn-taran-sahib', 'Tenkasi', 'Tezpur', 'Thalassery', 'Thanjavur', 'Theni', 'Thiruvalla',
'Thiruvallur', 'Thiruvananthapuram', 'Thiruvarur', 'Thodupuzha', 'Thoothukudi', 'Thoubal', 'Thrissur', 'Tindivanam', 'Tinsukia', 'Tiptur', 'Tirunelveli', 'Tirupati', 'Tirupur',
'Tiruttani', 'Tiruvannamalai', 'Tohana', 'Trichy', 'Tumakuru', 'Tuni', 'Udaipur', 'Udgir', 'Udhampur', 'Ujjain', 'Uluberia', 'Unnao', 'Uran-islampur', 'Vadodara', 'Valsad', 'Vapi',
'Varanasi', 'Varkala', 'Vellore', 'Veraval', 'Vidisha', 'Vijayawada', 'Viluppuram', 'Virudhunagar', 'Visnagar', 'Vizag', 'Vizianagaram', 'Vyara', 'Waidhan', 'Warangal', 'Wardha',
'Washim', 'Wayanad', 'Yamuna-nagar', 'Yavatmal'])
```

1. How many cities (including subregions) where Swiggy is having its restaurants listed?

```
list_of_cites_ind_sub=[]
for i in data.keys():
  if("restaurants" in list(data[i].keys())):
    list_of_cites_ind_sub.append(i)
  else:
    list_of_cites_ind_sub.append(i)
    for j in data[i]:
      if(j!='link'):
        list_of_cites_ind_sub.append(j)
```

```
len(list_of_cites_ind_sub)
```

```
998
```

```
len(set(list_of_cites_ind_sub))
```

```
995
```

There are 995 cities including the subregions

2 How many cities (don't include subregions) where Swiggy is having their restaurants listed?

```
list_of_cities=list(data.keys())
```

```
len(list_of_cities)
```

```
623
```

```
len(set(list_of_cities))
```

```
623
```

- There are 623 cities without subregions

- 3 The Subregion of Delhi with the maximum number of restaurants listed on Swiggy?

```
Delhi_subregion_no_rest={}
for i in data['Delhi'].keys():
  Delhi_subregion_no_rest[i]=len(data['Delhi'][i]['restaurants'].keys())
```

```
Delhi_subregion_no_rest
```

```
{'Tis Hazari': 0,
 'GTB Nagar': 685,
 'Jasola': 403,
 'Greater Noida': 5,
 'Mayur Vihar': 675,
 'BBK_MayurVihar': 1,
 'Laxmi Nagar': 933,
 'Punjabi Bagh': 186,
 'Old Delhi': 143,
 'Khan Market': 107,
 'Greater Kailash 2': 1038,
 'Ashok Vihar': 531,
 'Noida Expressway': 1,
 'South Extension': 378,
 'Kaushambi': 0,
 'Greater Kailash New': 81,
 'Vasant Kunj': 456,
 'Uttam Nagar': 528,
 'Dilshad Gardens': 677,
 'Connaught Place': 258,
 'South Campus': 329,
 'Rohini': 1138,
 'BBK_Chattarpur': 1,
 'Chattarpur': 636,
 'Lajpat Nagar': 449,
 'Kirti Nagar': 380,
 'Janakpuri': 0,
 'Karol Bagh': 0,
 'Golf Course': 0,
 'Indirapuram': 1279,
 'Dwarka': 722,
 'Rajouri Garden': 710,
 'NSP': 451,
 'Malviya Nagar': 901,
```

```
 'Sector 18': 3,
 'link': 301}
```

```
max_value = max(Delhi_subregion_no_rest.values())
max_keys = [k for k, v in Delhi_subregion_no_rest.items() if v == max_value]
```

```
print(max_keys,max_value)
```

```
['Indirapuram'] 1279
```

- Indirapuram is the subregion which has max num of resturant listed in Delhi

- 4 Name the top 5 Most Expensive Cities in the Datasets.

```
dict_values1={}
for i in data.keys():

    if("restaurants" in list(data[i].keys())):

        cost_of_each_rest=[]
        for rest_id in data[i]['restaurants'].keys():
            try:
                cost_val=int(data[i]['restaurants'][rest_id]['cost'].split()[1])
            except:
                cost_val=0
            cost_of_each_rest.append(cost_val)
            try:
                dict_values1[i]=sum(cost_of_each_rest)/len(cost_of_each_rest)
            except:
                dict_values1[i]=0

    else:

        cost_of_each_sub_cities=[]
        for sub_city in data[i].keys():
            cost_of_each_rest=[]
            for rest_id in data[i][sub_city]['restaurants'].keys():
                try:
                    cost_val=int(data[i][sub_city]['restaurants'][rest_id]['cost'].split()[1])
                except:
                    cost_val=0
                    pass
                cost_of_each_rest.append(cost_val)


            try:
```

```
                cost_of_each_sub_cities.append(sum(cost_of_each_rest)//len(cost_of_each_rest))
            except:

                cost_of_each_sub_cities.append(0)

        try:
            dict_values1[i]=sum(cost_of_each_sub_cities)/len(cost_of_each_sub_cities)
        except:
            dict_values1[i]=0
```

```
sorted_dict = sorted(dict_values1.items(), key=lambda item: -item[1])
```

```
sorted_dict[0:5]
```

```
[('Hinganghat', 18962.4375),
 ('South-goa', 469.401023890785),
 ('North-goa', 467.8957264957265),
 ('Gangtok', 466.6666666666667),
 ('Mussoorie', 460.0)]
```

▼ Here are the top 5 Expensive cities based on average cost of restaurants

▼ 5 List out the top 5 Restaurants with Maximum & minimum ratings throughout the dataset.

```
dict_values2={}
for i in data.keys():


    if("restaurants" in list(data[i].keys())):
        for rest_id in data[i]['restaurants'].keys():

            try:

                dict_values2[rest_id]=data[i]['restaurants'][rest_id]['rating']
            except:
                pass

    else:

        for sub_city in data[i].keys():
```

```
            for rest_id in data[i][sub_city]['restaurants'].keys():
                try:


                    dict_values2[rest_id]=data[i][sub_city]['restaurants'][rest_id]['rating']
                except:
                    pass
```

```
MAP_ID_NAME={}
for i in data.keys():


    if("restaurants" in list(data[i].keys())):
        for rest_id in data[i]['restaurants'].keys():

            try:

                MAP_ID_NAME[rest_id]=data[i]['restaurants'][rest_id]['name']
            except:
                pass

    else:

        for sub_city in data[i].keys():

            for rest_id in data[i][sub_city]['restaurants'].keys():
                try:


                    MAP_ID_NAME[rest_id]=data[i][sub_city]['restaurants'][rest_id]['name']
                except:
                    pass


# filtering and converting strings to floting numbers
filtered_dict={}
for i in dict_values2:
    if(dict_values2[i] != '--'):
        try:
            filtered_dict[i]=float(dict_values2[i])
        except:
            pass
```

```
sorted_dict_acc = sorted(filtered_dict.items(), key=lambda item: item[1])


#Top 5 minimum rating
sorted_dict_acc[0:5]
```

```
[('392000', 1.0),
 ('359391', 1.1),
 ('471181', 1.2),
 ('500567', 1.2),
 ('494551', 1.2)]
```

```
for i in sorted_dict_acc[0:5]:
    print(MAP_ID_NAME[i[0]],i[1])
```

```
 Ice Cream and Shakes Co 1.0
 Persian Delight 1.1
 SHAWARMA WRAP - ROLL YOUR SECRETS 1.2
 THE TARI STORY 1.2
 The Hyderabadi Biryani House 1.2
```

```
#Top 5 max rating
sorted_dict_acc[::-1][0:5]
```

```
[('498111', 5.0),
 ('498117', 5.0),
 ('498116', 5.0),
 ('563412', 5.0),
 ('563357', 5.0)]
```

```
for i in sorted_dict_acc[::-1][0:5]:
    print(MAP_ID_NAME[i[0]],i[1])
```

```
 DELHI GALLI 5.0
 KEBAB NATION 5.0
 HINDUSTANI BURGER 5.0
 Cream House 5.0
 Maa Ki Rasoi 5.0
```

▾ Above are the top 5 min and max rated restaurant id

▾ 6 Name of top 5 cities with the highest number of restaurants listed.

```
dict_values3={}
for i in data.keys():

    if("restaurants" in list(data[i].keys())):

        dict_values3[i]=len(data[i]['restaurants'].keys())


    else:
        sub_city1=[]
        for sub_city in data[i].keys():


            sub_city1.append(len(data[i][sub_city]['restaurants'].keys()))



        dict_values3[i]=sum(sub_city1)
```

```
sorted_dict_dec = sorted(dict_values3.items(), key=lambda item: -item[1])
```

```
sorted_dict_dec[0:5]
```

```
[('Bangalore', 16701),
 ('Delhi', 14386),
 ('Pune', 13145),
 ('Hyderabad', 12439),
 ('Chennai', 10987)]
```

Top 5 cities having highest num of restaurants listed

7 Top 10 cities as per the number of restaurants listed?

```
# This is the same question as above but with top 10
```

```
# Top 10 cites which has high count restaurants listed
sorted_dict_dec[0:10]
```

```
[('Bangalore', 16701),
 ('Delhi', 14386),
 ('Pune', 13145),
 ('Hyderabad', 12439),
 ('Chennai', 10987),
```

```
 ('Kolkata', 9651),
 ('Mumbai', 7221),
 ('Jaipur', 6250),
 ('Ahmedabad', 4736),
 ('Gurgaon', 4106)]
```

```
# Top 10 cites which has low count restaurants listed
sorted_dict_dec[::-1][0:10]
```

```
[('Wayanad', 0),
 ('Viluppuram', 0),
 ('Unnao', 0),
 ('Udhampur', 0),
 ('Tiruttani', 0),
 ('Tindivanam', 0),
 ('Thoubal', 0),
 ('Suri', 0),
 ('Sitamarhi', 0),
 ('Sirkali', 0)]
```

8 Name the top 5 Most Popular Restaurants in Pune.

popularity of a resturant is based on ratings

+ Code + Text

```
dict_values2={}
for i in data.keys():
    if(i=="Pune"):


        if("restaurants" in list(data[i].keys())):
            for rest_id in data[i]['restaurants'].keys():

                try:

                    dict_values2[rest_id]=data[i]['restaurants'][rest_id]['rating']
                except:
                    pass

        else:

            for sub_city in data[i].keys():

                for rest_id in data[i][sub_city]['restaurants'].keys():
                    try:


                        dict_values2[rest_id]=data[i][sub_city]['restaurants'][rest_id]['rating']
                    except:
```

```
                        pass


helper_dict={}
popular_rest={}

for i in dict_values2.keys():
    helper_dict[i]=MAP_ID_NAME[i]

list_helper=list(helper_dict.values())


for item in list_helper:
    popular_rest[item] = popular_rest.get(item, 0) + 1



sorted_dict_dec = sorted(popular_rest.items(), key=lambda item: -item[1])


sorted_dict_dec[:5]
```

```
[("Domino's Pizza", 26),
 ('Rolls Mania', 25),
 ('BOX8 - Desi Meals', 23),
 ('Kwality Walls Frozen Dessert and Ice Cream Shop', 22),
 ('ZAZA 22 Spice Biryani', 22)]
```

▼ Here are the top popular resturant in Pune

▼ 9 Which SubRegion in Delhi is having the least expensive restaurant in terms of cost?

```
d={}
for sub_city in data['Delhi'].keys():
    cost_of_each_rest=[]
    for rest_id in data['Delhi'][sub_city]['restaurants'].keys():



        try:
```

```
            cost_val=int(data['Delhi'][sub_city]['restaurants'][rest_id]['cost'].split()[1])

        except:
            cost_val=0
            pass
        cost_of_each_rest.append(cost_val)
    try:
        d[sub_city]=sum(cost_of_each_rest)/len(cost_of_each_rest)
    except:
        pass



sorted_dict_dec = sorted(d.items(), key=lambda item: item[1])


sorted_dict_dec[1:6]
```

```
[('Sector 18', 236.66666666666666),
 ('Uttam Nagar', 253.3219696969697),
 ('Laxmi Nagar', 270.1950696677385),
 ('Mayur Vihar', 275.05925925925925),
 ('Rohini', 277.6695957820738)]
```

▼ These are the sub cities which is least expensive restaurant and their avg price of cost

▼ 10 Top 5 most popular restaurant chains in India?

```
# Popularity of restaurant is based on how many different brances are there in across india


#We have a list of all restaurant names and ids
popular_rest={}
list_helper=list(MAP_ID_NAME.values())


for item in list_helper:
    popular_rest[item] = popular_rest.get(item, 0) + 1




sorted_dict_dec = sorted(popular_rest.items(), key=lambda item: -item[1])


sorted_dict_dec[:5]
```

```
[("Domino's Pizza", 468),
 ('Pizza Hut', 338),
 ('KFC', 325),
 ('Kwality Walls Frozen Dessert and Ice Cream Shop', 319),
 ('Baskin Robbins', 296)]
```

▼ These are the top popular restaurant chains in India

▼ 11 Which restaurant in Pune has the most number of people visiting?

▼ If the resturant has good num of ratings then we can consider as people has vitied more ,

this might not be true since we do not have people level info we are considring this.

```
dict_values2={}
for i in data.keys():
    if(i=="Pune"):


        if("restaurants" in list(data[i].keys())):
            for rest_id in data[i]['restaurants'].keys():

                try:

                    dict_values2[rest_id]=data[i]['restaurants'][rest_id]['rating']
                except:
                    pass

        else:

            for sub_city in data[i].keys():

                for rest_id in data[i][sub_city]['restaurants'].keys():
                    try:


                        dict_values2[rest_id]=data[i][sub_city]['restaurants'][rest_id]['rating']
                    except:
                        pass


filtered_dict={}
for i in dict_values2:
    if(dict_values2[i] != '--'):
        try:
            filtered_dict[i]=float(dict_values2[i])
```

```
        except:
            pass


sorted_dict_acc = sorted(filtered_dict.items(), key=lambda item: -item[1])


sorted_dict_acc[0:5]
```

```
[('533842', 5.0),
 ('558835', 5.0),
 ('572017', 5.0),
 ('534521', 5.0),
 ('558827', 5.0)]
```

```
for i in sorted_dict_acc[0:5]:
    print(MAP_ID_NAME[i[0]],i[1])
```

```
 The Samosa Man 5.0
 HRX by Eatfit 5.0
 CIRCLE OF CRUST 5.0
 Smoodies & Juices: Your Fruitfully 5.0
 HRX by Eatfit 5.0
```

12 Top 10 Restaurants with Maximum Ratings in Banglore

```
dict_values2={}
for i in data.keys():
    if(i=="Bangalore"):


        if("restaurants" in list(data[i].keys())):
            for rest_id in data[i]['restaurants'].keys():

                try:

                    dict_values2[rest_id]=data[i]['restaurants'][rest_id]['rating']
                except:
                    pass

        else:

            for sub_city in data[i].keys():

                for rest_id in data[i][sub_city]['restaurants'].keys():
                    try:
```

```
                    dict_values2[rest_id]=data[i][sub_city]['restaurants'][rest_id]['rating']
                except:
                    pass


filtered_dict={}
for i in dict_values2:
    if(dict_values2[i] != '--'):
        try:
            filtered_dict[i]=float(dict_values2[i])
        except:
            pass


sorted_dict_acc = sorted(filtered_dict.items(), key=lambda item: -item[1])


sorted_dict_acc[0:10]
```

```
[('528286', 5.0),
 ('530981', 5.0),
 ('530240', 5.0),
 ('557976', 5.0),
 ('447104', 5.0),
 ('570480', 5.0),
 ('558761', 5.0),
 ('560433', 5.0),
 ('536573', 5.0),
 ('561875', 5.0)]
```

```
for i in sorted_dict_acc[0:10]:
    print(MAP_ID_NAME[i[0]],i[1])
```

```
Creams and Bites 5.0
Bansal's Thali & Combo Meals 5.0
GOWHEY- HEALTHY DESSERT COMPANY 5.0
JUST CREAMERY - Artisanal Healthy Ice Cream 5.0
The Chinese Connection 5.0
SHAWARMA INDIAH 5.0
HRX by Eatfit 5.0
HUNGER TREATS 5.0
Snack Ride 5.0
JUST CREAMERY - Artisanal Healthy Ice Cream 5.0
```

▾ These are the top 10 rated restaurants in banglore

▾ 13 Top 10 Restaurant in Patna w.r.t rating

```
dict_values2={}
for i in data.keys():
    if(i=="Patna"):


        if("restaurants" in list(data[i].keys())):
            for rest_id in data[i]['restaurants'].keys():

                try:

                    dict_values2[rest_id]=data[i]['restaurants'][rest_id]['rating']
                except:
                    pass

        else:

            for sub_city in data[i].keys():

                for rest_id in data[i][sub_city]['restaurants'].keys():
                    try:


                        dict_values2[rest_id]=data[i][sub_city]['restaurants'][rest_id]['rating']
                    except:
                        pass


filtered_dict={}
for i in dict_values2:
    if(dict_values2[i] != '--'):
        try:
            filtered_dict[i]=float(dict_values2[i])
        except:
            pass


sorted_dict_acc = sorted(filtered_dict.items(), key=lambda item: -item[1])


sorted_dict_acc[0:10]
```

```
 [('290590', 4.6),
  ('550814', 4.6),
  ('561789', 4.5),
  ('550713', 4.5),
  ('553020', 4.5),
  ('560369', 4.5),
  ('561782', 4.4),
  ('550907', 4.4),
  ('492420', 4.4),
  ('469839', 4.4)]
```

```
for i in sorted_dict_acc[0:10]:
    print(MAP_ID_NAME[i[0]],i[1])
```

```
 Kwality Walls Frozen Dessert and Ice Cream Shop 4.6
 Slurpy Shakes 4.6
 Kouzina Kafe The Food Court 4.5
 Cupcake Bliss Cake & Desserts 4.5
 Burger It Up 4.5
 GARAGE KITCHEN 4.5
 Indiana Burgers 4.4
 The Dessert Zone 4.4
 Pure Veg Meals by Lunchbox 4.4
 KINGDOM OF MOMOS 4.4
```

These are the top 10 rated restaurants in Patna