PYTHON : Mini Project

---

1. Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

---

2. Set Options

In [3]:
```
pd.set_option("display.max_columns",40)
pd.set_option("display.max_rows",30)
pd.set_option("display.float_format",lambda x :f"{x:,.4f}")
```

---

3. Read Data

In [2]:
```
df = pd.read_csv("/content/drive/MyDrive/DATA SCIENCE AND ML AI/PYTHON/Project
df.head()
```

Out[2]:

	res_id	name	establishment	url	address	city	c
0	3400299	Bikanervala	Quick Bites	https://www.zomato.com/agra/bikanervala-khanda...	Kalyani Point, Near Tulsi Cinema, Bypass Road,...	Agra	
1	3400005	Mama Chicken Mama Franky House	Quick Bites	https://www.zomato.com/agra/mama-chicken-mama-...	Main Market, Sadar Bazaar, Agra Cantt, Agra	Agra	
2	3401013	Bhagat Halwai	Quick Bites	https://www.zomato.com/agra/bhagat-halwai-2-sh...	62/1, Near Easy Day, West Shivaji Nagar, Goalp...	Agra	
3	3400290	Bhagat Halwai	Quick Bites	https://www.zomato.com/agra/bhagat-halwai-civi...	Near Anjana Cinema, Nehru Nagar, Civil Lines, ...	Agra	
4	3401744	The Salt Cafe Kitchen & Bar	Casual Dining	https://www.zomato.com/agra/the-salt-cafe-kitc...	1C,3rd Floor, Fatehabad Road, Tajganj, Agra	Agra	

5 rows × 26 columns

4. Understand and Prepare the Data

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211944 entries, 0 to 211943
Data columns (total 26 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   res_id                211944 non-null  int64  
 1   name                  211944 non-null  object 
 2   establishment         207117 non-null  object 
 3   url                   211944 non-null  object 
 4   address               211810 non-null  object 
 5   city                  211944 non-null  object 
 6   city_id               211944 non-null  int64  
 7   locality              211944 non-null  object 
 8   latitude              211944 non-null  float64
 9   longitude             211944 non-null  float64
 10  zipcode               48757 non-null   object 
 11  country_id            211944 non-null  int64  
 12  locality_verbose      211944 non-null  object 
 13  cuisines              210553 non-null  object 
 14  timings               208070 non-null  object 
 15  average_cost_for_two  211944 non-null  int64  
 16  price_range           211944 non-null  int64  
 17  currency              211944 non-null  object 
 18  highlights            209875 non-null  object 
 19  aggregate_rating      211944 non-null  float64
 20  rating_text           211944 non-null  object 
 21  votes                 211944 non-null  int64  
 22  photo_count           211944 non-null  int64  
 23  opentable_support     211896 non-null  float64
 24  delivery              211944 non-null  int64  
 25  takeaway              211944 non-null  int64  
dtypes: float64(4), int64(9), object(13)
memory usage: 42.0+ MB
```

5. Understand the variables

In [5]:
```
df.describe()
```

Out[5]:

	res_id	city_id	latitude	longitude	country_id	av
count	211,944.0000	211,944.0000	211,944.0000	211,944.0000	211,944.0000	
mean	13,494,112.3481	4,746.7854	21.4995	77.6153	1.0000	
std	7,883,721.9725	5,568.7664	22.7813	7.5001	0.0000	
min	50.0000	1.0000	0.0000	0.0000	1.0000	
25%	3,301,027.0000	11.0000	15.4961	74.8780	1.0000	
50%	18,695,734.0000	34.0000	22.5142	77.4260	1.0000	
75%	18,812,974.0000	11,306.0000	26.8412	80.2193	1.0000	
max	19,159,790.0000	11,354.0000	10,000.0000	91.8328	1.0000	

In [6]:
```
df.dtypes
```

Out[6]:

	0
res_id	int64
name	object
establishment	object
url	object
address	object
city	object
city_id	int64
locality	object
latitude	float64
longitude	float64
zipcode	object
country_id	int64
locality_verbose	object
cuisines	object
timings	object
average_cost_for_two	int64
price_range	int64
currency	object
highlights	object
aggregate_rating	float64
rating_text	object
votes	int64
photo_count	int64
opentable_support	float64
delivery	int64
takeaway	int64

dtype: object

6. Check for Missing Values

In [7]:
```
df.isnull().sum()
```

Out[7]:

	0
res_id	0
name	0
establishment	4827
url	0
address	134
city	0
city_id	0
locality	0
latitude	0
longitude	0
zipcode	163187
country_id	0
locality_verbose	0
cuisines	1391
timings	3874
average_cost_for_two	0
price_range	0
currency	0
highlights	2069
aggregate_rating	0
rating_text	0
votes	0
photo_count	0
opentable_support	48
delivery	0
takeaway	0

dtype: int64

In [8]:
```
df.isnull().mean()
```

Out[8]:

	0
res_id	0.0000
name	0.0000
establishment	0.0228
url	0.0000
address	0.0006
city	0.0000
city_id	0.0000
locality	0.0000
latitude	0.0000
longitude	0.0000
zipcode	0.7700
country_id	0.0000
locality_verbose	0.0000
cuisines	0.0066
timings	0.0183
average_cost_for_two	0.0000
price_range	0.0000
currency	0.0000
highlights	0.0098
aggregate_rating	0.0000
rating_text	0.0000
votes	0.0000
photo_count	0.0000
opentable_support	0.0002
delivery	0.0000
takeaway	0.0000

dtype: float64

As i analyze the data ..'opentable_support' column havin all the values in 0 .. so we can also drop this column

In [9]:
```
df = df.drop('opentable_support', axis=1)
```

In [10]:
```
df
```

Out[10]:

	res_id	name	establishment	url	address
0	3400299	Bikanervala	Quick Bites	https://www.zomato.com/agra/bikanervala-khanda...	Kalyani Point, Near Tulsi Cinema, Bypass Road,...
1	3400005	Mama Chicken Mama Franky House	Quick Bites	https://www.zomato.com/agra/mama-chicken-mama-...	Main Market, Sadar Bazaar, Agra Cantt, Agra
2	3401013	Bhagat Halwai	Quick Bites	https://www.zomato.com/agra/bhagat-halwai-2-sh...	62/1, Near Easy Day, West Shivaji Nagar, Goalp...
3	3400290	Bhagat Halwai	Quick Bites	https://www.zomato.com/agra/bhagat-halwai-civi...	Near Anjana Cinema, Nehru Nagar, Civil Lines, ...
4	3401744	The Salt Cafe Kitchen & Bar	Casual Dining	https://www.zomato.com/agra/the-salt-cafe-kitc...	1C,3rd Floor, Fatehabad Road, Tajganj, Agra
...	...	...	...	...	...
211939	3202251	Kali Mirch Cafe And Restaurant	Casual Dining	https://www.zomato.com/vadodara/kali-mirch-caf...	Manu Smriti Complex, Near Navrachna School, GI...
211940	3200996	Raju Omlet	Quick Bites	https://www.zomato.com/vadodara/raju-omlet-kar...	Mahalaxmi Apartment, Opposite B O B, Karoli Ba...
211941	18984164	The Grand Thakar	Casual Dining	https://www.zomato.com/vadodara/the-grand-thak...	3rd Floor, Shreem Shalini

	res_id	name	establishment	url	address
					Mall, Opposite Conqu...
211942	3201138	Subway	Quick Bites	https://www.zomato.com/vadodara/subway-1-akota...	G-2, Vedant Platina, Near Cosmos, Akota, Vadodara
211943	18879846	Freshcos - The Health Cafe	Café	https://www.zomato.com/vadodara/freshcos-the-h...	Shop 7, Ground Floor, Opposite Natubhai Circle...

211944 rows × 25 columns

*# -- As we can see values in zipcode column mostly null (77%)..so we can drop this column *

In [11]:
```
df.drop('zipcode', axis=1, inplace=True)
```

In [12]:
```
df.isna().sum()
```

Out[12]:

	0
res_id	0
name	0
establishment	4827
url	0
address	134
city	0
city_id	0
locality	0
latitude	0
longitude	0
country_id	0
locality_verbose	0
cuisines	1391
timings	3874
average_cost_for_two	0
price_range	0
currency	0
highlights	2069
aggregate_rating	0
rating_text	0
votes	0
photo_count	0
delivery	0
takeaway	0

dtype: int64

Now as we can see in the column 'esatablishment' some values are missing .. we can fill at those missing values by mode (this column has object data type)

In [13]:
```
df['establishment']=df['establishment'].fillna(df['establishment'].mode()[0])
```

In [14]:
```
df.isna().mean()
```

Out[14]:

	0
res_id	0.0000
name	0.0000
establishment	0.0000
url	0.0000
address	0.0006
city	0.0000
city_id	0.0000
locality	0.0000
latitude	0.0000
longitude	0.0000
country_id	0.0000
locality_verbose	0.0000
cuisines	0.0066
timings	0.0183
average_cost_for_two	0.0000
price_range	0.0000
currency	0.0000
highlights	0.0098
aggregate_rating	0.0000
rating_text	0.0000
votes	0.0000
photo_count	0.0000
delivery	0.0000
takeaway	0.0000

dtype: float64

In [15]:
```
df.dtypes
```

Out[15]:

	0
res_id	int64
name	object
establishment	object
url	object
address	object
city	object
city_id	int64
locality	object
latitude	float64
longitude	float64
country_id	int64
locality_verbose	object
cuisines	object
timings	object
average_cost_for_two	int64
price_range	int64
currency	object
highlights	object
aggregate_rating	float64
rating_text	object
votes	int64
photo_count	int64
delivery	int64
takeaway	int64

dtype: object

*here all the columns('cuisines','timing','highlights'), have some missing values having object data type .. so i am applying same mode method to fill the values same as before *

In [16]:
```
df['cuisines']=df['cuisines'].fillna(df['cuisines'].mode()[0])
```

In [17]:
```
df['timings']=df['timings'].fillna(df['timings'].mode()[0])
```

In [18]:
```
df['highlights']=df['highlights'].fillna(df['highlights'].mode()[0])
```

In [19]:
```
df.isna().mean()
```

Out[19]:

	0
res_id	0.0000
name	0.0000
establishment	0.0000
url	0.0000
address	0.0006
city	0.0000
city_id	0.0000
locality	0.0000
latitude	0.0000
longitude	0.0000
country_id	0.0000
locality_verbose	0.0000
cuisines	0.0000
timings	0.0000
average_cost_for_two	0.0000
price_range	0.0000
currency	0.0000
highlights	0.0000
aggregate_rating	0.0000
rating_text	0.0000
votes	0.0000
photo_count	0.0000
delivery	0.0000
takeaway	0.0000

dtype: float64

*and for the address i just fill the value as in the city column *

In [20]:
```
df['address'] = df['address'].fillna(df['city'])
```

In [21]:
```
df
```

Out[21]:

	res_id	name	establishment	url	address
0	3400299	Bikanervala	Quick Bites	https://www.zomato.com/agra/bikanervala-khanda...	Kalyani Point, Near Tulsi Cinema, Bypass Road,...
1	3400005	Mama Chicken Mama Franky House	Quick Bites	https://www.zomato.com/agra/mama-chicken-mama-...	Main Market, Sadar Bazaar, Agra Cantt, Agra
2	3401013	Bhagat Halwai	Quick Bites	https://www.zomato.com/agra/bhagat-halwai-2-sh...	62/1, Near Easy Day, West Shivaji Nagar, Goalp...
3	3400290	Bhagat Halwai	Quick Bites	https://www.zomato.com/agra/bhagat-halwai-civi...	Near Anjana Cinema, Nehru Nagar, Civil Lines, ...
4	3401744	The Salt Cafe Kitchen & Bar	Casual Dining	https://www.zomato.com/agra/the-salt-cafe-kitc...	1C,3rd Floor, Fatehabad Road, Tajganj, Agra
...	...	...	...	...	...
211939	3202251	Kali Mirch Cafe And Restaurant	Casual Dining	https://www.zomato.com/vadodara/kali-mirch-caf...	Manu Smriti Complex, Near Navrachna School, GI...
211940	3200996	Raju Omlet	Quick Bites	https://www.zomato.com/vadodara/raju-omlet-kar...	Mahalaxmi Apartment, Opposite B O B, Karoli Ba...
211941	18984164	The Grand Thakar	Casual Dining	https://www.zomato.com/vadodara/the-grand-thak...	3rd Floor, Shreem Shalini

	res_id	name	establishment	url	address
					Mall, Opposite Conqu...
211942	3201138	Subway	Quick Bites	https://www.zomato.com/vadodara/subway-1-akota...	G-2, Vedant Platina, Near Cosmos, Akota, Vadodara
211943	18879846	Freshcos - The Health Cafe	Café	https://www.zomato.com/vadodara/freshcos-the-h...	Shop 7, Ground Floor, Opposite Natubhai Circle...

211944 rows × 24 columns

In [22]:
```
df.isna().sum()
```

Out[22]:

	0
res_id	0
name	0
establishment	0
url	0
address	0
city	0
city_id	0
locality	0
latitude	0
longitude	0
country_id	0
locality_verbose	0
cuisines	0
timings	0
average_cost_for_two	0
price_range	0
currency	0
highlights	0
aggregate_rating	0
rating_text	0
votes	0
photo_count	0
delivery	0
takeaway	0

dtype: int64

---

7. Study Correlation

In [23]:
```
num_columns = ['average_cost_for_two','price_range','aggregate_rating','votes'
corr = df[num_columns].corr()
```

```
corr
```

Out[23]:

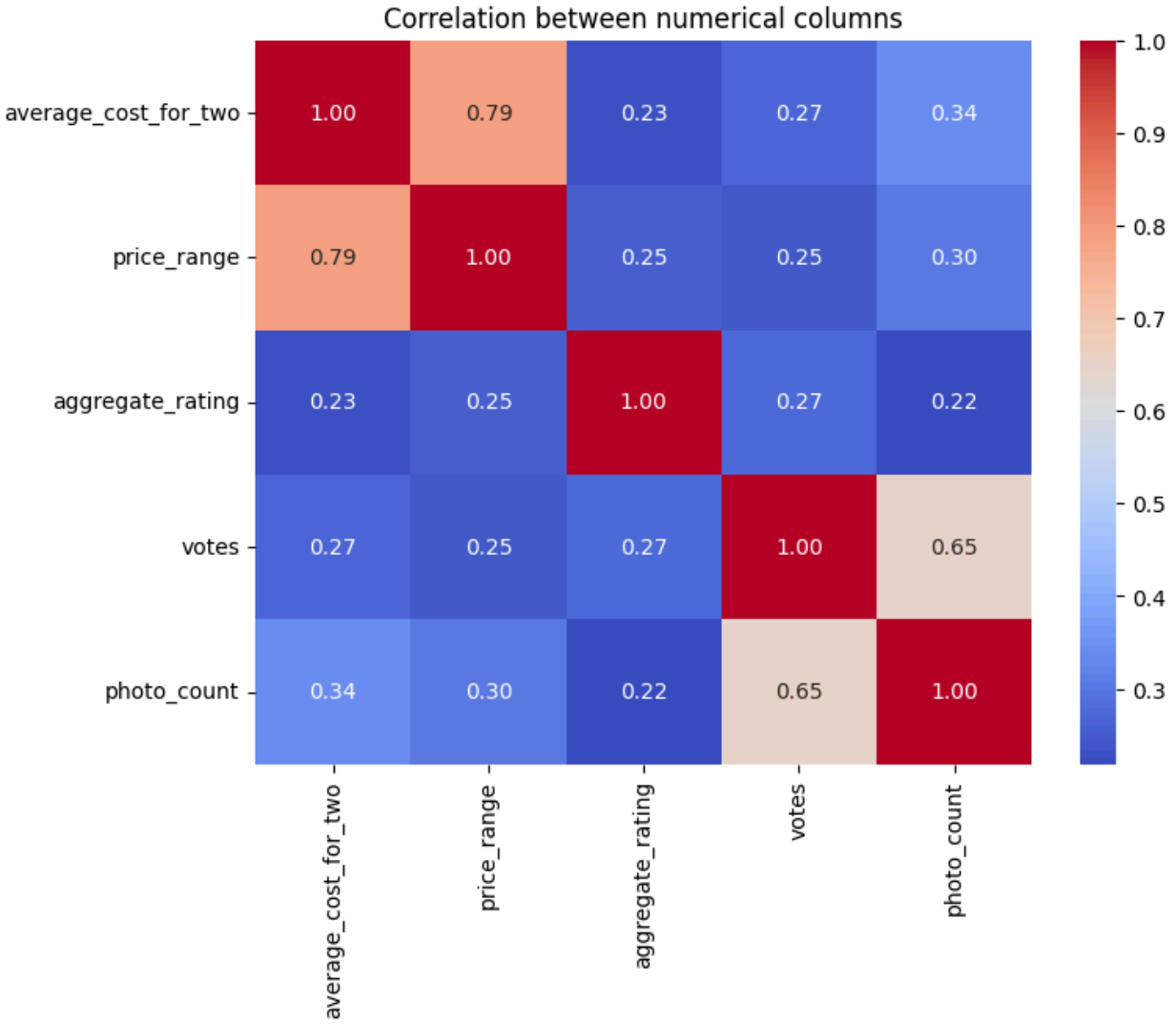
	average_cost_for_two	price_range	aggregate_rating	vot
average_cost_for_two	1.0000	0.7859	0.2323	0.26
price_range	0.7859	1.0000	0.2523	0.24
aggregate_rating	0.2323	0.2523	1.0000	0.27
votes	0.2690	0.2450	0.2720	1.00
photo_count	0.3419	0.3014	0.2185	0.65

In [ ]:
```
## Outcomes:

# average_cost_for_two <-> price_range (0.7859): it shows strong positive -> h
# average_cost_for_two <-> aggregate_rating(0.2323) : Weak positive -> expensi
# average_cost_for_two <-> photo_count(0.3419) : Moderate positive -> expensiv
# aggregate_rating <-> votes(0.2720) : Weak positive -> more votes usually mea
# aggregate_rating <-> price_range(0.2523) : Weak positive -> premium restaura
# votes <-> photo_count(0.6550) : Strong positive -> more photos uploaded usua
```

In [24]:
```
# Creating Heatmap:

plt.figure(figsize=(8,6))
sns.heatmap(corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation between numerical columns")
plt.show()
```



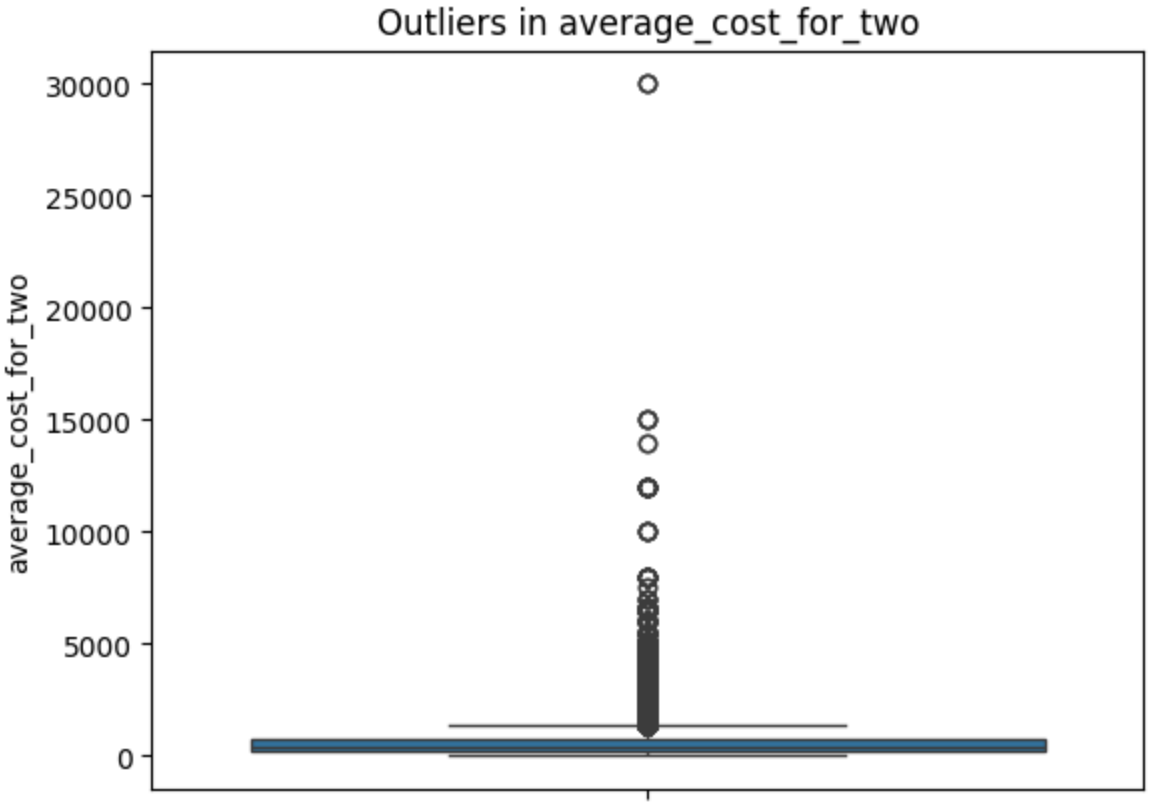
---

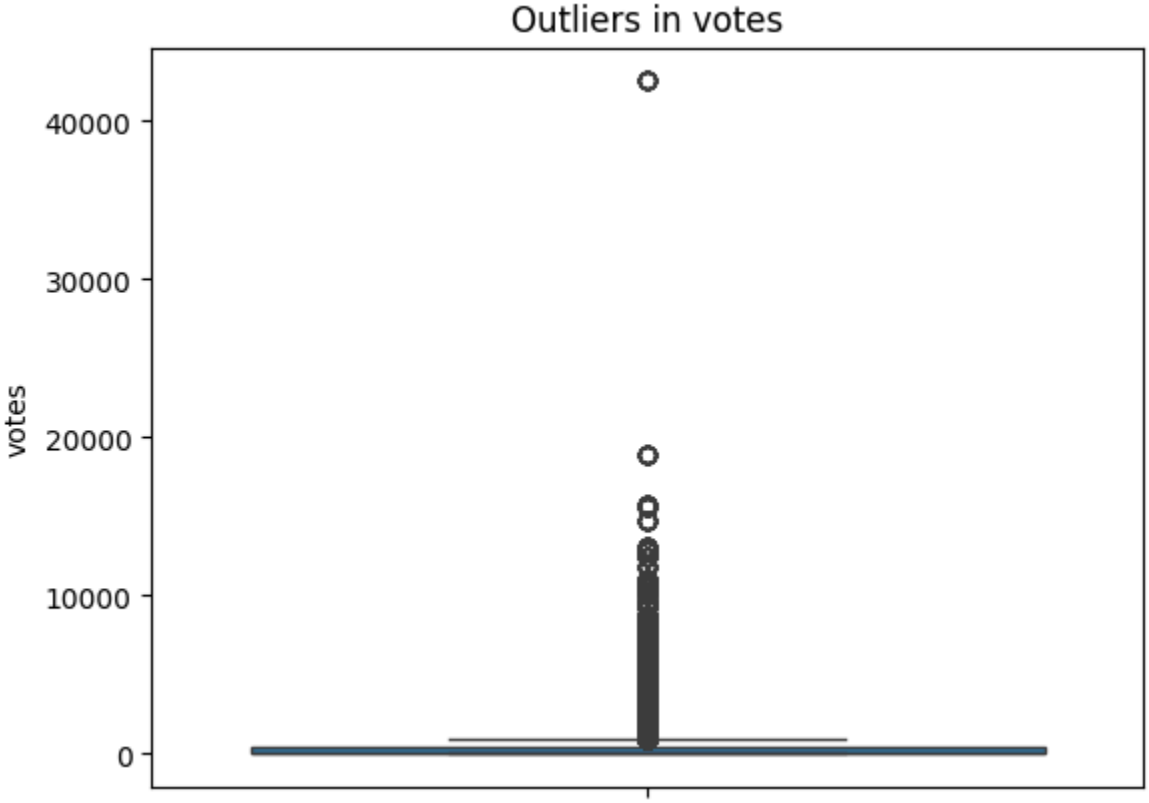
8. Detect Outliers

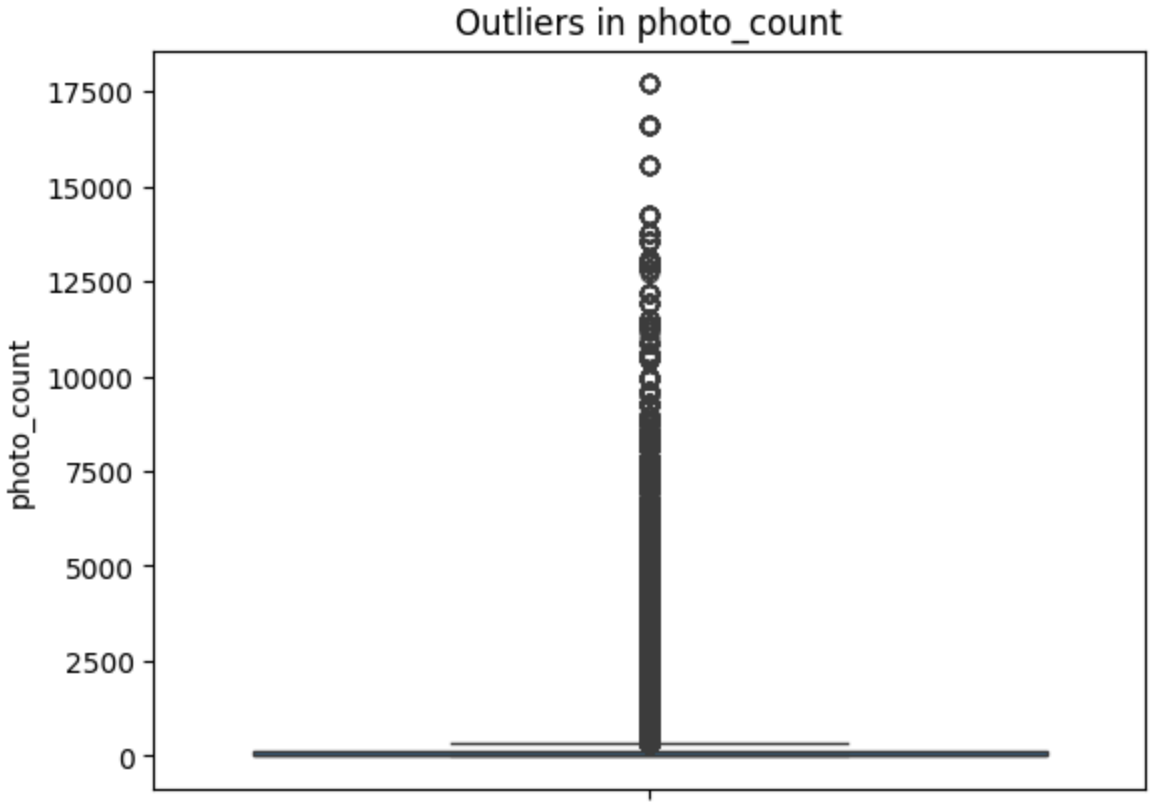
In [36]:
```
sns.boxplot(y=df['average_cost_for_two'])
plt.title("Outliers in average_cost_for_two")
plt.show()

sns.boxplot(y=df['votes'])
plt.title("Outliers in votes")
plt.show()

sns.boxplot(y=df['photo_count'])
plt.title("Outliers in photo_count")
plt.show()
```


Outliers in average_cost_for_two
average_cost_for_two
30000
25000
20000
15000
10000
5000
0


Outliers in votes
votes
40000
30000
20000
10000
0



In [ ]:
```
# --> As i can see average_cost_for_two columns has one very high outlier (₹30
# --> As i can see votes columns has one very high outlier (above 40000) and s
# --> As i can see photo_count has 3 big outlier(above 15000)
```

In [ ]:
```

```

---

9. Creating a new column 'region'

In [25]:
```
df['city'].unique()
```

Out[25]:
```
array(['Agra', 'Ahmedabad', 'Gandhinagar', 'Ajmer', 'Alappuzha',
       'Allahabad', 'Amravati', 'Amritsar', 'Aurangabad', 'Bangalore',
       'Bhopal', 'Bhubaneshwar', 'Chandigarh', 'Mohali', 'Panchkula',
       'Zirakpur', 'Nayagaon', 'Chennai', 'Coimbatore', 'Cuttack',
       'Darjeeling', 'Dehradun', 'New Delhi', 'Gurgaon', 'Noida',
       'Faridabad', 'Ghaziabad', 'Greater Noida', 'Dharamshala',
       'Gangtok', 'Goa', 'Gorakhpur', 'Guntur', 'Guwahati', 'Gwalior',
       'Haridwar', 'Hyderabad', 'Secunderabad', 'Indore', 'Jabalpur',
       'Jaipur', 'Jalandhar', 'Jammu', 'Jamnagar', 'Jamshedpur', 'Jhansi',
       'Jodhpur', 'Junagadh', 'Kanpur', 'Kharagpur', 'Kochi', 'Kolhapur',
       'Kolkata', 'Howrah', 'Kota', 'Lucknow', 'Ludhiana', 'Madurai',
       'Manali', 'Mangalore', 'Manipal', 'Udupi', 'Meerut', 'Mumbai',
       'Thane', 'Navi Mumbai', 'Mussoorie', 'Mysore', 'Nagpur',
       'Nainital', 'Nashik', 'Neemrana', 'Ooty', 'Palakkad', 'Patiala',
       'Patna', 'Puducherry', 'Pune', 'Pushkar', 'Raipur', 'Rajkot',
       'Ranchi', 'Rishikesh', 'Salem', 'Shimla', 'Siliguri', 'Srinagar',
       'Surat', 'Thrissur', 'Tirupati', 'Trichy', 'Trivandrum', 'Udaipur',
       'Varanasi', 'Vellore', 'Vijayawada', 'Vizag', 'Vadodara'],
      dtype=object)
```

In [26]:
```
# Defining some cities :
city_region = {
    'Agra': 'North-East',
    'Delhi': 'North', 'New Delhi': 'North', 'Gurgaon': 'North', 'Noida': 'Nort
    'Mumbai': 'West', 'Pune': 'West',
    'Bangalore': 'South', 'Chennai': 'South', 'Hyderabad': 'South',
    'Kolkata': 'East', 'Patna': 'East',
    'Bhopal': 'Central', 'Indore': 'Central'

}

# Apply mapping, using map
df['region'] = df['city'].map(city_region)

# Replace NaN with "Other"
df['region'] = df['region'].fillna('Other')

print(df[['city','region']].head(25))
```

	city	region
0	Agra	North-East
1	Agra	North-East
2	Agra	North-East
3	Agra	North-East
4	Agra	North-East
5	Agra	North-East
6	Agra	North-East
7	Agra	North-East
8	Agra	North-East
9	Agra	North-East
10	Agra	North-East
11	Agra	North-East
12	Agra	North-East
13	Agra	North-East
14	Agra	North-East
15	Agra	North-East
16	Agra	North-East
17	Agra	North-East
18	Agra	North-East
19	Agra	North-East
20	Agra	North-East
21	Agra	North-East
22	Agra	North-East
23	Agra	North-East
24	Agra	North-East

In [27]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211944 entries, 0 to 211943
Data columns (total 25 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   res_id                211944 non-null  int64  
 1   name                  211944 non-null  object 
 2   establishment         211944 non-null  object 
 3   url                   211944 non-null  object 
 4   address               211944 non-null  object 
 5   city                  211944 non-null  object 
 6   city_id               211944 non-null  int64  
 7   locality              211944 non-null  object 
 8   latitude              211944 non-null  float64
 9   longitude             211944 non-null  float64
 10  country_id            211944 non-null  int64  
 11  locality_verbose      211944 non-null  object 
 12  cuisines              211944 non-null  object 
 13  timings               211944 non-null  object 
 14  average_cost_for_two  211944 non-null  int64  
 15  price_range           211944 non-null  int64  
 16  currency              211944 non-null  object 
 17  highlights            211944 non-null  object 
 18  aggregate_rating      211944 non-null  float64
 19  rating_text           211944 non-null  object 
 20  votes                 211944 non-null  int64  
 21  photo_count           211944 non-null  int64  
 22  delivery              211944 non-null  int64  
 23  takeaway              211944 non-null  int64  
 24  region                211944 non-null  object 
dtypes: float64(3), int64(9), object(13)
memory usage: 40.4+ MB
```

In [28]:
```
df.isna().sum()
```

Out[28]:

	0
res_id	0
name	0
establishment	0
url	0
address	0
city	0
city_id	0
locality	0
latitude	0
longitude	0
country_id	0
locality_verbose	0
cuisines	0
timings	0
average_cost_for_two	0
price_range	0
currency	0
highlights	0
aggregate_rating	0
rating_text	0
votes	0
photo_count	0
delivery	0
takeaway	0
region	0

dtype: int64

In [ ]: