9	<pre>import pandas as pd import numpy as np import matplotlib.pyplot as plt %matplotlib inline import seaborn as sns df=pd.read_csv('zomato.csv.zip',encoding= 'latin-1')</pre>
Out[4]:	Restaurant ID Restaurant Code Code City Address Locality Verbose Longitude Latitude Cuisines Currency Has Table booking delivery Price order name of the Control of the Control of the Control of Century City Mall, Poblacion, Makati City Malk Poblacion, Makati City Mak
	1 6304287 Izakaya Kikufuji 162 Makati City Chino Roces Avenue, Legaspi Edsa Shangri-La, S
	2 6300002 Edsa Shangri-La 162 Mandaluyong Shangri-La 162 Mandaluyong City City City City Ortigas, Mandaluyong Ortigas, Mandaluyong City Ma Third Hall, SM Megamall, O Third Megamall, O Th
5	4 6314302 Sambo 162 Mandaluyong Kojin 162 Mandaluyong City SM Megamall, Ortigas, Mandaluyong City Megamall, Ortigas Signatural City Megamall, Ortigas Signatural City Megamall, Ortigas Signatural City Megamall, Ortigas Signatural City Mandal
out[5]: []]	<pre>df.columns Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',</pre>
it[19]: (In [6]: (In	(9551, 21) df.info() <class 'pandas.core.frame.dataframe'=""> RangeIndex: 9551 entries, 0 to 9550 Data columns (total 21 columns): # Column Non-Null Count Dtype</class>
	0 Restaurant ID 9551 non-null object 1 Restaurant Name 9551 non-null int64 2 Country Code 9551 non-null object 3 City 9551 non-null object 4 Address 9551 non-null object 5 Locality 9551 non-null object 6 Locality Verbose 9551 non-null float64 7 Longitude 9551 non-null float64 8 Latitude 9551 non-null object 9 Cuisines 9542 non-null object 10 Average Cost for two 9551 non-null int64
	11 Currency 9551 non-null object 12 Has Table booking 9551 non-null object 13 Has Online delivery 9551 non-null object 14 Is delivering now 9551 non-null object 15 Switch to order menu 9551 non-null object 16 Price range 9551 non-null int64 17 Aggregate rating 9551 non-null float64 18 Rating color 9551 non-null object 19 Rating text 9551 non-null object 20 Votes 9551 non-null int64 dtypes: float64(3), int64(5), object(13)
In [8]: (memory usage: 1.5+ MB df.describe() Restaurant ID Country Code Longitude Latitude Average Cost for two Price range Aggregate rating Votes count 9.551000e+03 9551.000000 9551.00000 9551.000000 9551.000000 9551.000000 9551.000000 9551.000000 9551.00000 9551.00000 9551.000000 9551.000000 9551.0000
	25% 3.019625e+05 1.000000 77.081343 28.478713 250.000000 1.000000 2.500000 5.000000 50% 6.004089e+06 1.000000 77.191964 28.570469 400.000000 2.000000 31.000000 75% 1.835229e+07 1.000000 77.282006 28.642758 700.000000 2.000000 3.700000 131.000000 max 1.850065e+07 216.000000 174.832089 55.976980 80000.000000 4.000000 4.900000 10934.000000 Data Analysis
1 [10]: (ut[10]: F	1.Misssing Values 2.explore about the numerical features 3.explore about the categorical features 4.Finding relationship between features df.isnull().sum() Restaurant ID 0 Restaurant Name 0 Country Code 0 City 0 Address 0
L L C <i>A</i> C H	Locality Verbose 0 Locality Verbose 0 Longitude 0 Latitude 0 Cuisines 9 Average Cost for two 0 Currency 0 Has Table booking 0 Has Online delivery 0 Is delivering now 0 Switch to order menu 0
F F F V C	Switch to order menu
	sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis') <axessubplot:> ===</axessubplot:>
	Restaurant ID - Restaurant Name - Country Code - Country Code - Country Code - Address - Locality - Locality - Locality Verbose - Lorgitude - Latitude - Currency - Has Table booking - Has Table booking - Rating color - Rating color - Rating color - Rating text - Votes -
n [16]: (df_country=pd.read_excel('Country-Code.xlsx') df_country.head() Country Code
2 3 4 1 [20]: 1	1 14 Australia 2 30 Brazil 3 37 Canada 4 94 Indonesia final_df=pd.merge(df,df_country,on='Country Code',how='left') final_df.head()
ut[21]: _	Restaurant ID Restaurant Name Country Code City Address Locality Locality Verbose Longitude Latitude Cuisines Has Table Online delivery only only only only only only only onl
	1 6304287 Izakaya Kikufuji 162 Makati City Little Tokyo, Legaspi Village, Roces Avenue, Legaspi Legaspi Village, Makati City
;	Ortigas, Mandal Third Floor, Megamall, Ortigas, Megamall, Ortigas, Mandaluyong City Megamall, Ortigas, Mandaluyong Megamall, Ortigas, Mandaluyong City Megamall, Ortigas, Mandaluyong City Mandaluyong Megamall, Ortigas, Mandaluyong City, Mandal Third Floor, Megamall, Ortigas, Mandaluyong City, Mandal Third Floor, Megamall, Ortigas, Mandaluyong City, Mandal Third Megamall, Megamal
5 n [22]: [1	
L L C C C C C C C C C C C C C C C C C C	Restaurant Name object Country Code int64 City object Address object Locality object Locality Verbose object Longitude float64 Catitude float64 Cuisines object Average Cost for two object Currency object
C H H S F F F F	Currency object Has Table booking object Has Online delivery object Is delivering now object Switch to order menu object Price range int64 Aggregate rating float64 Rating color object Rating text object Votes int64 Country object
n [32]: (an [36]: (an [36]: 1	<pre>dtype: object country_names=final_df.Country.value_counts().index country_names Index(['India', 'United States', 'United Kingdom', 'Brazil', 'UAE',</pre>
n [34]: (4 ut[34]: 6	<pre>country_values=final_df.Country.value_counts().values country_values array([8652, 434, 80, 60, 60, 60, 40, 34, 24, 22, 21, 20, 20, 20, 4], dtype=int64) #top 3 countries plt.pie(country_values[:3], labels=country_names[:3], autopct='%1.1f%%') ([<matplotlib.patches.wedge 0x22efaf9a730="" at="">,</matplotlib.patches.wedge></pre>
ut[48]: ⁽	([<matplotlib.patches.wedge 0x22efaf9a730="" at="">,</matplotlib.patches.wedge>
I	India 94.4% United Kingdom United States
	<pre>final_df.columns Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',</pre>
n [50]: 1 ut[50]:	<pre>dtype='object') ratings=final_df.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index().rename(columns={0:'Rating Count'})) ratings Aggregate rating Rating color Rating text Rating Count 0 0.0 White Not rated 2148 1 1.8 Red Poor 1</pre>
	2 1.9 Red Poor 2 3 2.0 Red Poor 7 4 2.1 Red Poor 15 5 2.2 Red Poor 27 6 2.3 Red Poor 47 7 2.4 Red Poor 87 8 2.5 Orange Average 110
1 1 1 1 1 1 1	9 2.6 Orange Average 191 10 2.7 Orange Average 250 11 2.8 Orange Average 315 12 2.9 Orange Average 381 13 3.0 Orange Average 468 14 3.1 Orange Average 519 15 3.2 Orange Average 522
1 1 1 2 2	16 3.3 Orange Average 483 17 3.4 Orange Average 498 18 3.5 Yellow Good 480 19 3.6 Yellow Good 458 20 3.7 Yellow Good 427 21 3.8 Yellow Good 400 22 3.9 Yellow Good 335
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	22 3.9 Yellow Good 335 23 4.0 Green Very Good 266 24 4.1 Green Very Good 274 25 4.2 Green Very Good 221 26 4.3 Green Very Good 174 27 4.4 Green Very Good 144 28 4.5 Dark Green Excellent 95 29 4.6 Dark Green Excellent 78
3 3 n [54]: i	30 4.7 Dark Green Excellent 42 31 4.8 Dark Green Excellent 25 32 4.9 Dark Green Excellent 61 import matplotlib matplotlib matplotlib.rcParams['figure.figsize'] =(12,6) sns.barplot(x='Aggregate rating', y='Rating Count', data=ratings) <axessubplot:xlabel='aggregate rating',="" ylabel="Rating Count"></axessubplot:xlabel='aggregate>
ut[54].	2000 -
	1000 - Superior of the state of
	Où 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 Aggregate rating final_df.columns Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
ut[61]:	'Has Online delivery', 'Is delivering now', 'Switch to order menu', 'Price range', 'Aggregate rating', 'Rating color', 'Rating text', 'Votes', 'Country'], dtype='object') final_df.groupby(['Aggregate rating', 'Country']).size().reset_index().head() Aggregate rating
3 2 1 [63]: 1 1t[63]:	2
	0Botswana Pula(P)Phillipines221Brazilian Real(R\$)Brazil602Dollar(\$)Australia243Dollar(\$)Canada44Dollar(\$)Singapore205Dollar(\$)United States4346Emirati Diram(AED)UAE60
1 1	6 Emirati Diram(AED) UAE 60 7 Indian Rupees(Rs.) India 8652 8 Indonesian Rupiah(IDR) Indonesia 21 9 NewZealand(\$) New Zealand 40 10 Pounds(⊞) United Kingdom 80 11 Qatari Rial(QR) Qatar 20 12 Rand(R) South Africa 60 13 Sri Lankan Rupee(LKR) Sri Lanka 20
1 n [64]: 1 ut[64]: _	14 Turkish Lira(TL) Turkey 34 final_df.groupby(['Has Online delivery', 'Country']).size().reset_index() Has Online delivery Country 0 No Australia 24 1 No Brazil 60
	2 No Canada 4 3 No India 6229 4 No Indonesia 21 5 No New Zealand 40 6 No Phillipines 22 7 No Qatar 20 8 No Singapore 20
1 1 1 1 1	9 No South Africa 60 10 No Sri Lanka 20 11 No Turkey 34 12 No United Kingdom 80 14 No United States 434 15 Yes India 2423
1 [65]: (c) 1 [66]: (c)	<pre>16 Yes UAE 28 city_names=final_df.City.value_counts().index city_names Index(['New Delhi', 'Gurgaon', 'Noida', 'Faridabad', 'Ghaziabad', 'Bhubaneshwar', 'Amritsar', 'Ahmedabad', 'Lucknow', 'Guwahati',</pre>
n [68]:	'Ojo Caliente', 'Montville', 'Monroe', 'Miller', 'Middleton Beach',
	20, 2
	1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
	Text(0.8789045225625368, -0.6614581167535246, 'Noida'), Text(1.0922218418223437, -0.13058119407559224, 'Faridabad'), Text(1.099946280005612, -0.010871113182029924, 'Ghaziabad')], [Text(-0.3352010631374145, 0.497634652402289, '68.9%'), Text(0.0340186500653484, -0.5990348332507311, '14.1%'), Text(0.47940246685229276, -0.36079533641101336, '13.6%'), Text(0.5957573682667329, -0.07122610585941394, '3.2%'), Text(0.5999706981848791, -0.005929698099289049, '0.3%')]) New Delhi
	68.9% Ghaziabad Faridabad 13.6%
	13.6% 14.1% Noida
	Gurgaon Cuisines_val=final_df.Cuisines.value_counts().values
[71]: (interpretation of the content	Cuisines_val=final_df.Cuisines.value_counts().values Cuisines_val array([936, 511, 354,, 1, 1, 1], dtype=int64) Cuisines_names=final_df.Cuisines.value_counts().index Cuisines_names Index(['North Indian', 'North Indian, Chinese', 'Chinese', 'Fast Food',
[71]: (at [71]: 6 [72]: (at [73]: (a	Cuisines_val=final_df.Cuisines.value_counts().values Cuisines_val array([936, 511, 354,, 1, 1, 1], dtype=int64) Cuisines_names=final_df.Cuisines.value_counts().index Cuisines_names Index(['North Indian', 'North Indian, Chinese', 'Chinese', 'Fast Food',
[71]: (1) it[71]: (2) it[71]: (3) it[73]: (4) it[73]: (7)	Cuisines_val=final_df.Cuisines.value_counts().values Cuisines_val array([936, 511, 354,, 1, 1, 1], dtype=int64) Cuisines_names=final_df.Cuisines.value_counts().index Cuisines_names Index(['North Indian, 'North Indian, Chinese', 'Chinese', 'Fast Food',
[71]: (1) t[71]: (2) t[71]: (3) t[72]: (4) t[73]: (7)	Cuisines_val=final_df.Cuisines.value_counts().values Cuisines_val array([936, 511, 354,, 1, 1, 1], dtype=int64) Cuisines_names=final_df.Cuisines.value_counts().index Cuisines_names Index(['North Indian, 'North Indian, Chinese', 'Chinese', 'Fast Food',
[71]: (1) t[71]: (2) t[72]: (3) t[73]: (4) t[73]: (7)	Cutsines val=final df.Cutsines.value_counts().values Cutsines_val= rarq([386, 511, 354,, 1, 1, 1], dtype=in(04) Cutsines_names=final df.Cutsines_value_counts().index Cutsines_names Index(["North Indian", 'North Indian, Chinese', 'Chinese', 'East Food',
[71]: (1) t[71]: (2) t[71]: (3) t[72]: (4) t[73]: (7) t[75]: (7)	Cuisines_val=final_df.Cuisines.value_counts().values Cuisines_val Cray(1885, 511, 354,, 1, 1, 1], drype=int64) Cuisines names=final df.Cuisines.value counts() index North Indian, Nuplai, 'Cairé, 'Bakery', 'North Indian, Nuplai, 'Cairé, 'Bakery', 'Street Food', 'Coan, American, Portuguese', 'South Indian, Desertis, Everages', 'Healthy food, footh Indian, Italian, Salad, 'Bengali, Fast Food', 'North Indian, Norld Cuisines, 'Asian', 'Bengali, Fast Food', 'North Indian, Norld Cuisines, 'Asian', 'Bengali, South Indian', 'Titalian, Norld Cuisines, 'Asian', Bengali, Titalian, Chinese'), 'Titalian, Titalian, 'Asian', Bengali, 'Titalian, 'Titalian, 'Titalian, 'Titalian, 'Titalian, 'Titalian, '