**COURSE CODE: DJS22ITL405** **DATE:**29/04/2024

**COURSE NAME:** Programing with Python Laboratory **CLASS:** SYBTECH

**NAME:** Jugal Pandav **SAP:** 60003220239

**BATCH**: I2-1  **Roll No**.:I070

# EXPERIMENT NO. 12

**CO/LO: CO1,CO2,CO3**

# AIM / OBJECTIVE: Mini-Project

# PROBLEM STATEMENT:Data Analysis Of Zomato Dataset

# DATASET DESCRIPTION: We used a Zomato dataset and used the python libraries pandas , numpy and matplotlib to do data analysis and plot appropriate graphs.

# IMPLEMENTATION:

**Code :**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import matplotlib

matplotlib.rcParams['figure.figsize'] = (20,12)

plt.rcParams["figure.autolayout"] = True

df = pd.read\_csv('zomato.csv',encoding='latin-1')

print(df.shape) #Printing number of rows and columns

print(df.head()) # Printing first 5 rows of dataset

print(df.columns) #Printing list of all columns

print(df.info()) # Gives information about data type of rows , and null count of each column

print(df.describe()) #Provides statistical information about each numerical column , like standard deviation , mean , quartiles , etc.

"""## In Data Analysis What All Things We Can Do

1) Count of Missing Values , Replace Missing Values and Delete Missing Values

2) Explore the Numerical Variables

3) Explore About Categorical Variables

4) Finding Relationship Between Features

"""

print(df.isnull().sum()) #Printing number of null values in each column

print([features for features in df.columns if df[features].isnull().sum()>0])  #Printing columns with > 0 null values

plt.bar(df.columns  , df.isnull().sum())

plt.xlabel("Columns")

plt.ylabel("Null Count")

plt.xticks(rotation = 45)

plt.show()

df\_country = pd.read\_excel('Country-Code.xlsx') # There are additional columns , containing country codes of each country

print(df\_country.head())

final\_df = pd.merge(df,df\_country,on="Country Code",how="left") # Adding a Country Code column to dataframe , using merge method of pandas

print(final\_df.head())

print([f for f in final\_df.columns if final\_df[f].isnull().sum()>0]) # Checking if Country Code has any null values

print(df['Country Code'].dtype) # Checking data type of country code

print(final\_df.columns) # These are the final columns in df

print(final\_df.Country.value\_counts()) #India has highest ratings , then US and UK

final\_df\_numec = final\_df.select\_dtypes(exclude='object') # Using select\_dtypes to exclude non numerical values

print(final\_df\_numec.corr()) #We can see correlation between each column

Country\_name = final\_df.Country.value\_counts().index # This is list of country names

Country\_val = final\_df.Country.value\_counts().values # These are the values associated with each country

plt.pie(Country\_val[:3],labels=Country\_name[:3],autopct='%.2f%%') #Pie chart showing percentage of ratings of top three countries

plt.show()

"""# Observation From Above Pie

Zomato has it maximum records and transaction with India, then US and then UK

"""

#Making a separate ratings dataframe using groupby function

ratings = final\_df.groupby(['Aggregate rating','Rating color','Rating text']).size().reset\_index().rename(columns={0:'Rating count'})

print(ratings)

"""## Observation

Ratings Range and text

4.5 - 4.9 => Excellent, Dark Green

4.0 - 4.4 => Very Good, Green

3.5 - 4.9 => Good, Yellow

2.5 - 3.0 => Average, Orange

0.1 - 2.4 => Poor, Red

0     => Not Rated , White

"""

print(ratings.head())

plt.bar(ratings['Aggregate rating'] , ratings['Rating count'] , width=0.07 , color = 'blue') #We decrease width so that each bar could be visible , otherwise they were overlapping

plt.xlabel("Aggregate rating")

plt.ylabel("Rating count")

plt.show()

#So we can see that most ratings are 0 , and other ratings are following a normal distribution

"""#Observations#

##1) Most of the people has not rated

##2) maximum rating is between 2.5-3.4 i.e average

"""

rating\_color\_count = final\_df.groupby(['Rating color']).size().reset\_index().rename(columns = {0: "Count"}) #Making a new df which shows count of each rating colour

print(rating\_color\_count)

colors = ['darkgreen' , 'green' , 'orange' , 'red' , 'white' , 'yellow']

plt.bar(rating\_color\_count['Rating color'] , rating\_color\_count['Count'] , color = colors , edgecolor = 'black') #Using bar graph for the same graph as above

plt.show()

#So we can see that most reviews are Orange , i.e , [2.5,3.0]

plt.hist(final\_df['Rating color'], color = 'blue') #We can use histogram for categorical data also

plt.show()

"""#Finding out which country has given zero ratings"""

#Finding out which country has given zero rating

country\_with\_zero\_rating\_count = final\_df[final\_df['Rating color']=='White'].groupby(['Country']).size().reset\_index().rename(columns={0:'Country count'})

print(country\_with\_zero\_rating\_count)

plt.bar(country\_with\_zero\_rating\_count['Country'] , country\_with\_zero\_rating\_count['Country count'] , color = 'red')

plt.xlabel("Countries")

plt.ylabel("Count")

plt.show()

#So we can see that India has given most zero ratings

"""Observations

1) Maximum No. of people from India has not rated

#Finding out which currency is used in which country?

"""

#find out which currency is used in which country?

currency\_of\_countries = final\_df.groupby(['Country','Currency']).size().reset\_index()

print(currency\_of\_countries.columns)

currency\_of\_countries = currency\_of\_countries.drop(0,axis=1)

print(currency\_of\_countries) #So we get to know currency of each country

"""## Which Countries Have Online Delivery Options"""

online\_delivery\_countries = final\_df.groupby(['Country','Has Online delivery']).size().reset\_index()

print(online\_delivery\_countries)

availiblity\_online\_delivery\_option = online\_delivery\_countries.drop(online\_delivery\_countries.loc[online\_delivery\_countries['Has Online delivery']=='No'].index)

availiblity\_online\_delivery\_option = availiblity\_online\_delivery\_option.reset\_index()

availiblity\_online\_delivery\_option = availiblity\_online\_delivery\_option.drop('index',axis=1)

availiblity\_online\_delivery\_option = availiblity\_online\_delivery\_option.drop(0,axis=1)

print(availiblity\_online\_delivery\_option)

print(final\_df[final\_df['Has Online delivery']=='Yes'].Country.value\_counts())

"""#Observations :

#Online Delivery Options are available in India and UAE

"""

#Create a pie chart same as countries for city

cities\_name = final\_df.City.value\_counts().index

cities\_val = final\_df.City.value\_counts().values

plt.pie(cities\_val[:5],labels=cities\_name[:5],autopct='%.2f%%')

plt.show()

"""## Finding top 10 food items"""

cuisines\_count = final\_df.groupby(['Cuisines']).size().reset\_index().rename(columns={0:'Cuisines count'})

print(cuisines\_count)

cuisines\_count = cuisines\_count.sort\_values(by='Cuisines count',ascending=False).reset\_index().drop('index',axis=1)

cuisines\_count.head(10)

plt.bar(cuisines\_count['Cuisines'][:10] , cuisines\_count['Cuisines count'][:10] , color = 'blue')

plt.xticks(rotation=45)

plt.show()

india\_df = final\_df[final\_df['Country'] == 'India']

print(india\_df)

plt.hist(india\_df['Average Cost for two'], bins=10, color='green')

plt.show()

# So we can see that most meals have average cost for two between Rs.0-1000

**Output:**

**(9551, 21)**

**Restaurant ID Restaurant Name Country Code ... Rating color Rating text Votes**

**0 6317637 Le Petit Souffle 162 ... Dark Green Excellent 314**

**1 6304287 Izakaya Kikufuji 162 ... Dark Green Excellent 591**

**2 6300002 Heat - Edsa Shangri-La 162 ... Green Very Good 270**

**3 6318506 Ooma 162 ... Dark Green Excellent 365**

**4 6314302 Sambo Kojin 162 ... Dark Green Excellent 229**

**[5 rows x 21 columns]**

**Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',**

**'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',**

**'Average Cost for two', 'Currency', 'Has Table booking',**

**'Has Online delivery', 'Is delivering now', 'Switch to order menu',**

**'Price range', 'Aggregate rating', 'Rating color', 'Rating text',**

**'Votes'],**

**dtype='object')**

**<class 'pandas.core.frame.DataFrame'>**

**RangeIndex: 9551 entries, 0 to 9550**

**Data columns (total 21 columns):**

**# Column Non-Null Count Dtype**

**--- ------ -------------- -----**

**0 Restaurant ID 9551 non-null int64**

**1 Restaurant Name 9551 non-null object**

**2 Country Code 9551 non-null int64**

**3 City 9551 non-null object**

**4 Address 9551 non-null object**

**5 Locality 9551 non-null object**

**6 Locality Verbose 9551 non-null object**

**7 Longitude 9551 non-null float64**

**8 Latitude 9551 non-null float64**

**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)**

**memory usage: 1.5+ MB**

**None**

**Restaurant ID Country Code Longitude ... Price range Aggregate rating Votes**

**count 9.551000e+03 9551.000000 9551.000000 ... 9551.000000 9551.000000 9551.000000**

**mean 9.051128e+06 18.365616 64.126574 ... 1.804837 2.666370 156.909748**

**std 8.791521e+06 56.750546 41.467058 ... 0.905609 1.516378 430.169145**

**min 5.300000e+01 1.000000 -157.948486 ... 1.000000 0.000000 0.000000**

**25% 3.019625e+05 1.000000 77.081343 ... 1.000000 2.500000 5.000000**

**50% 6.004089e+06 1.000000 77.191964 ... 2.000000 3.200000 31.000000**

**75% 1.835229e+07 1.000000 77.282006 ... 2.000000 3.700000 131.000000**

**max 1.850065e+07 216.000000 174.832089 ... 4.000000 4.900000 10934.000000**

**[8 rows x 8 columns]**

**Restaurant ID 0**

**Restaurant Name 0**

**Country Code 0**

**City 0**

**Address 0**

**Locality 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**

**Price range 0**

**Aggregate rating 0**

**Rating color 0**

**Rating text 0**

**Votes 0**

**dtype: int64**

**['Cuisines']**

**Country Code Country**

**0 1 India**

**1 14 Australia**

**2 30 Brazil**

**3 37 Canada**

**4 94 Indonesia**

**Restaurant ID Restaurant Name Country Code ... Rating text Votes Country**

**0 6317637 Le Petit Souffle 162 ... Excellent 314 Phillipines**

**1 6304287 Izakaya Kikufuji 162 ... Excellent 591 Phillipines**

**2 6300002 Heat - Edsa Shangri-La 162 ... Very Good 270 Phillipines**

**3 6318506 Ooma 162 ... Excellent 365 Phillipines**

**4 6314302 Sambo Kojin 162 ... Excellent 229 Phillipines**

**[5 rows x 22 columns]**

**['Cuisines']**

**int64**

**Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',**

**'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',**

**'Average Cost for two', 'Currency', 'Has Table booking',**

**'Has Online delivery', 'Is delivering now', 'Switch to order menu',**

**'Price range', 'Aggregate rating', 'Rating color', 'Rating text',**

**'Votes', 'Country'],**

**dtype='object')**

**India 8652**

**United States 434**

**United Kingdom 80**

**Brazil 60**

**UAE 60**

**South Africa 60**

**New Zealand 40**

**Turkey 34**

**Australia 24**

**Phillipines 22**

**Indonesia 21**

**Singapore 20**

**Qatar 20**

**Sri Lanka 20**

**Canada 4**

**Name: Country, dtype: int64**

**Restaurant ID Country Code Longitude ... Price range Aggregate rating Votes**

**Restaurant ID 1.000000 0.148471 -0.226081 ... -0.134540 -0.326212 -0.147023**

**Country Code 0.148471 1.000000 -0.698299 ... 0.243327 0.282189 0.154530**

**Longitude -0.226081 -0.698299 1.000000 ... -0.078939 -0.116818 -0.085101**

**Latitude -0.052081 0.019792 0.043207 ... -0.166688 0.000516 -0.022962**

**Average Cost for two -0.001693 0.043225 0.045891 ... 0.075083 0.051792 0.067783**

**Price range -0.134540 0.243327 -0.078939 ... 1.000000 0.437944 0.309444**

**Aggregate rating -0.326212 0.282189 -0.116818 ... 0.437944 1.000000 0.313691**

**Votes -0.147023 0.154530 -0.085101 ... 0.309444 0.313691 1.000000**

**[8 rows x 8 columns]**

**Aggregate rating Rating color Rating text Rating count**

**0 0.0 White Not rated 2148**

**1 1.8 Red Poor 1**

**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**

**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**

**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**

**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**

**30 4.7 Dark Green Excellent 42**

**31 4.8 Dark Green Excellent 25**

**32 4.9 Dark Green Excellent 61**

**Aggregate rating Rating color Rating text Rating count**

**0 0.0 White Not rated 2148**

**1 1.8 Red Poor 1**

**2 1.9 Red Poor 2**

**3 2.0 Red Poor 7**

**4 2.1 Red Poor 15**

**Rating color Count**

**0 Dark Green 301**

**1 Green 1079**

**2 Orange 3737**

**3 Red 186**

**4 White 2148**

**5 Yellow 2100**

**Country Country count**

**0 Brazil 5**

**1 India 2139**

**2 United Kingdom 1**

**3 United States 3**

**Index(['Country', 'Currency', 0], dtype='object')**

**Country Currency**

**0 Australia Dollar($)**

**1 Brazil Brazilian Real(R$)**

**2 Canada Dollar($)**

**3 India Indian Rupees(Rs.)**

**4 Indonesia Indonesian Rupiah(IDR)**

**5 New Zealand NewZealand($)**

**6 Phillipines Botswana Pula(P)**

**7 Qatar Qatari Rial(QR)**

**8 Singapore Dollar($)**

**9 South Africa Rand(R)**

**10 Sri Lanka Sri Lankan Rupee(LKR)**

**11 Turkey Turkish Lira(TL)**

**12 UAE Emirati Diram(AED)**

**13 United Kingdom Pounds(£)**

**14 United States Dollar($)**

**Country Has Online delivery 0**

**0 Australia No 24**

**1 Brazil No 60**

**2 Canada No 4**

**3 India No 6229**

**4 India Yes 2423**

**5 Indonesia No 21**

**6 New Zealand No 40**

**7 Phillipines No 22**

**8 Qatar No 20**

**9 Singapore No 20**

**10 South Africa No 60**

**11 Sri Lanka No 20**

**12 Turkey No 34**

**13 UAE No 32**

**14 UAE Yes 28**

**15 United Kingdom No 80**

**16 United States No 434**

**Country Has Online delivery**

**0 India Yes**

**1 UAE Yes**

**India 2423**

**UAE 28**

**Name: Country, dtype: int64**

**Cuisines Cuisines count**

**0 Afghani 4**

**1 Afghani, Mughlai, Chinese 1**

**2 Afghani, North Indian 1**

**3 Afghani, North Indian, Pakistani, Arabian 1**

**4 African 1**

**... ... ...**

**1 ... Good 71 India**

**626 3400005 Time2Eat - Mama Chicken 1 ... Good 94 India**

**627 3400021 Chokho Jeeman Marwari Jain Bhojanalya 1 ... Very Good 87 India**

**628 3400017 Pinch Of Spice 1 ... Very Good 177 India**

**... ... ... ... ... ... ... ...**

**9271 2800100 D Cabana 1 ... Good 193 India**

**9272 2800418 Kaloreez 1 ... Good 85 India**

**9273 2800881 Plot 17 1 ... Very Good 172 India**

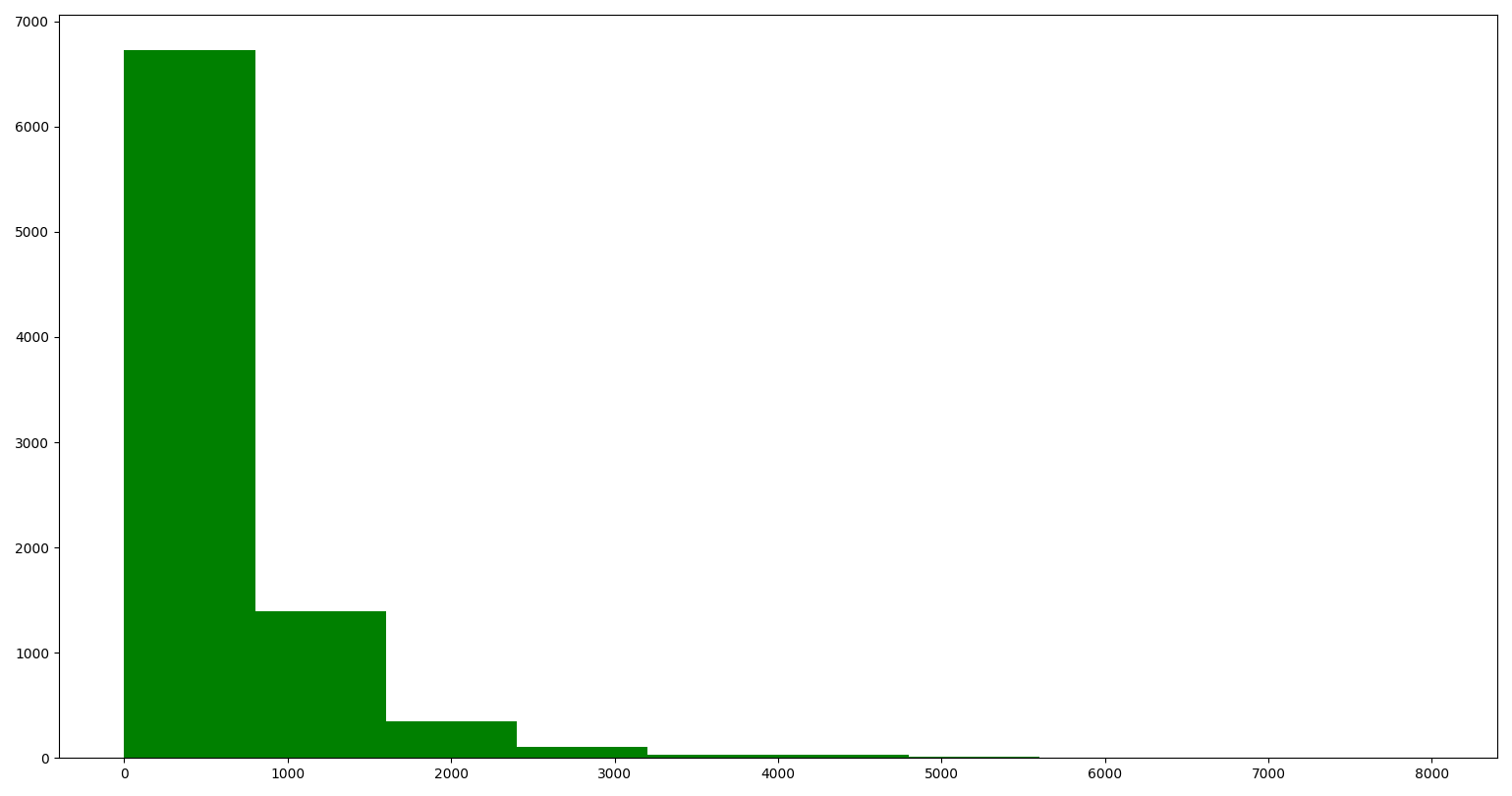
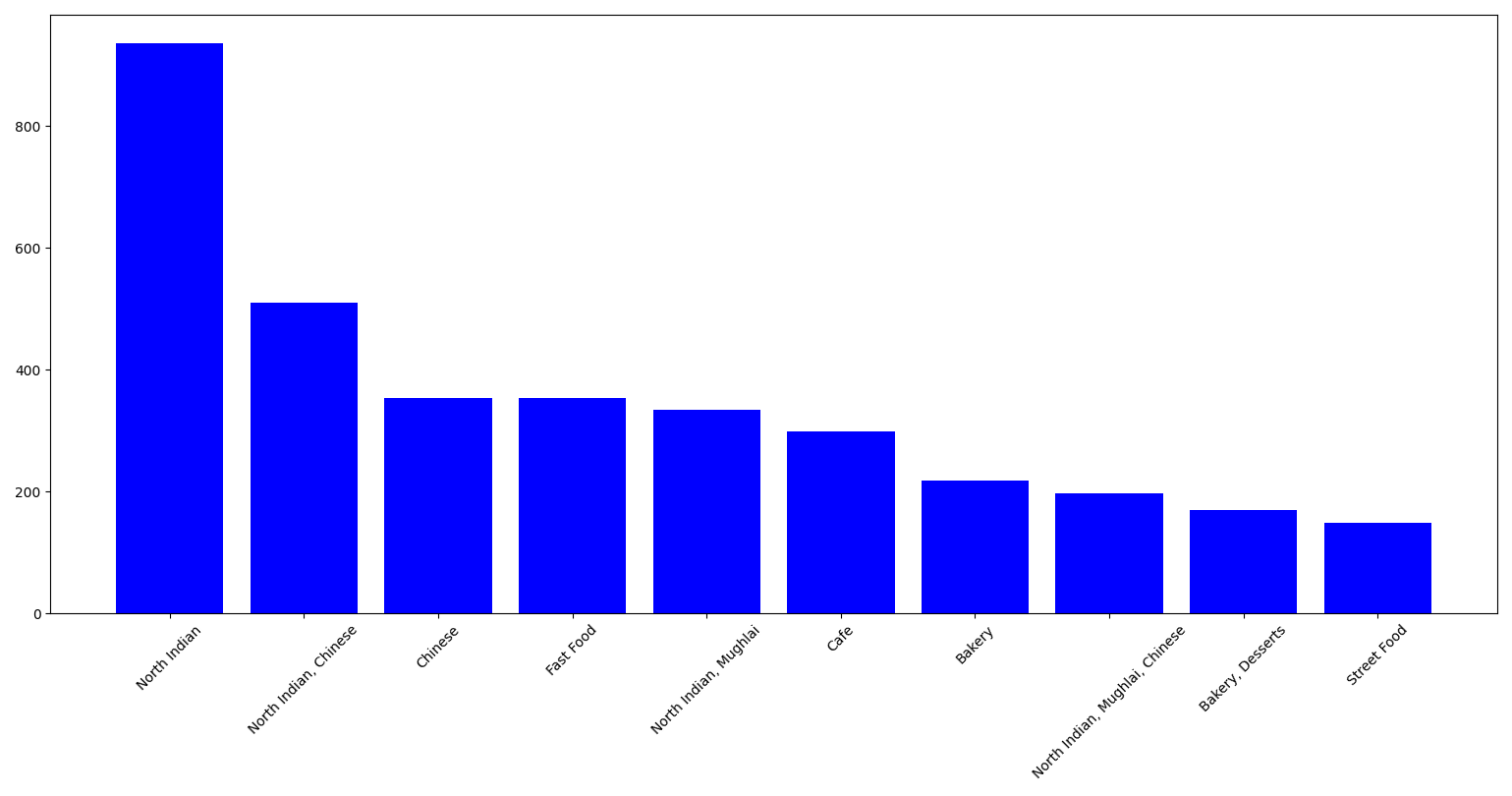
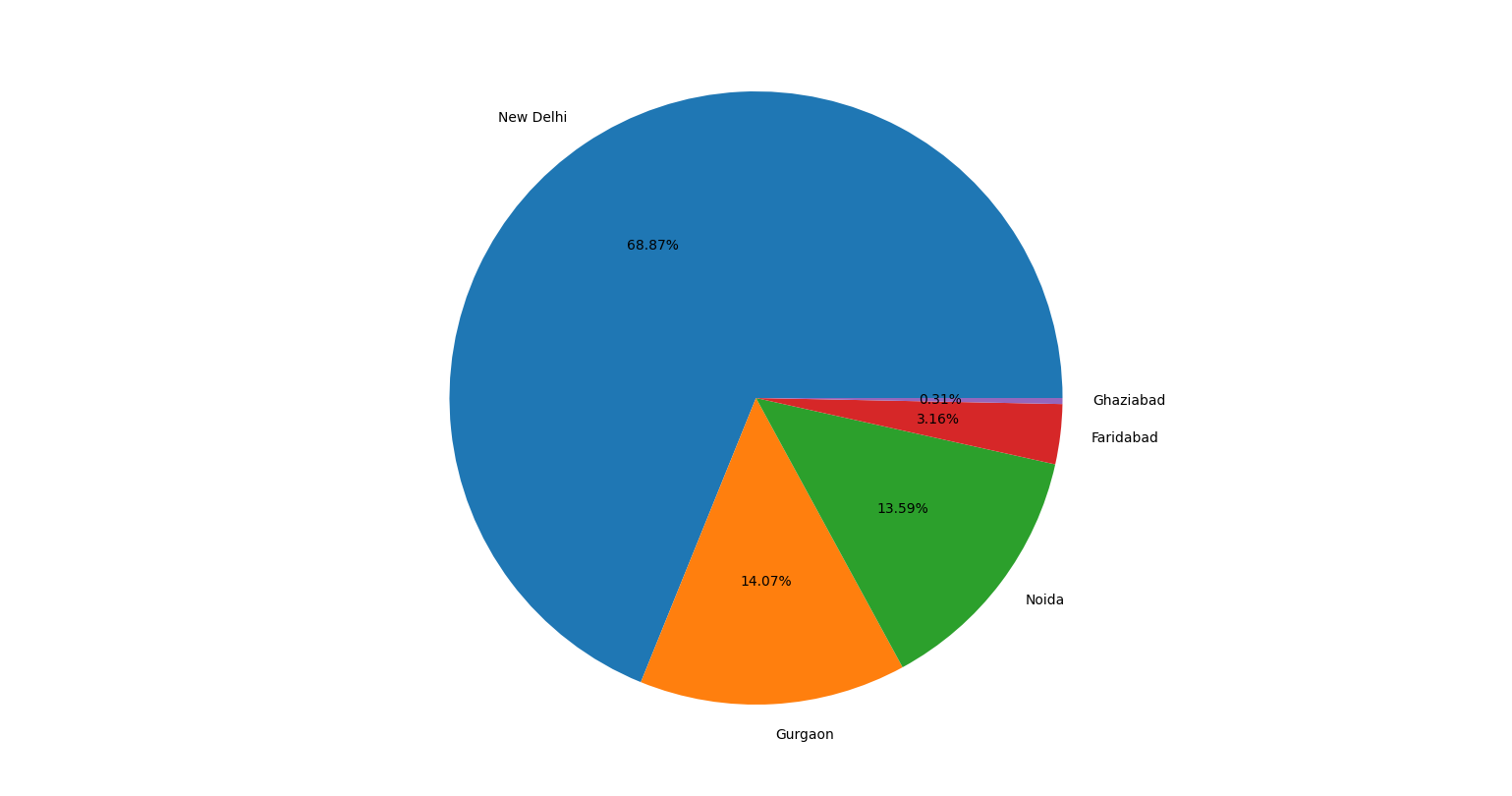
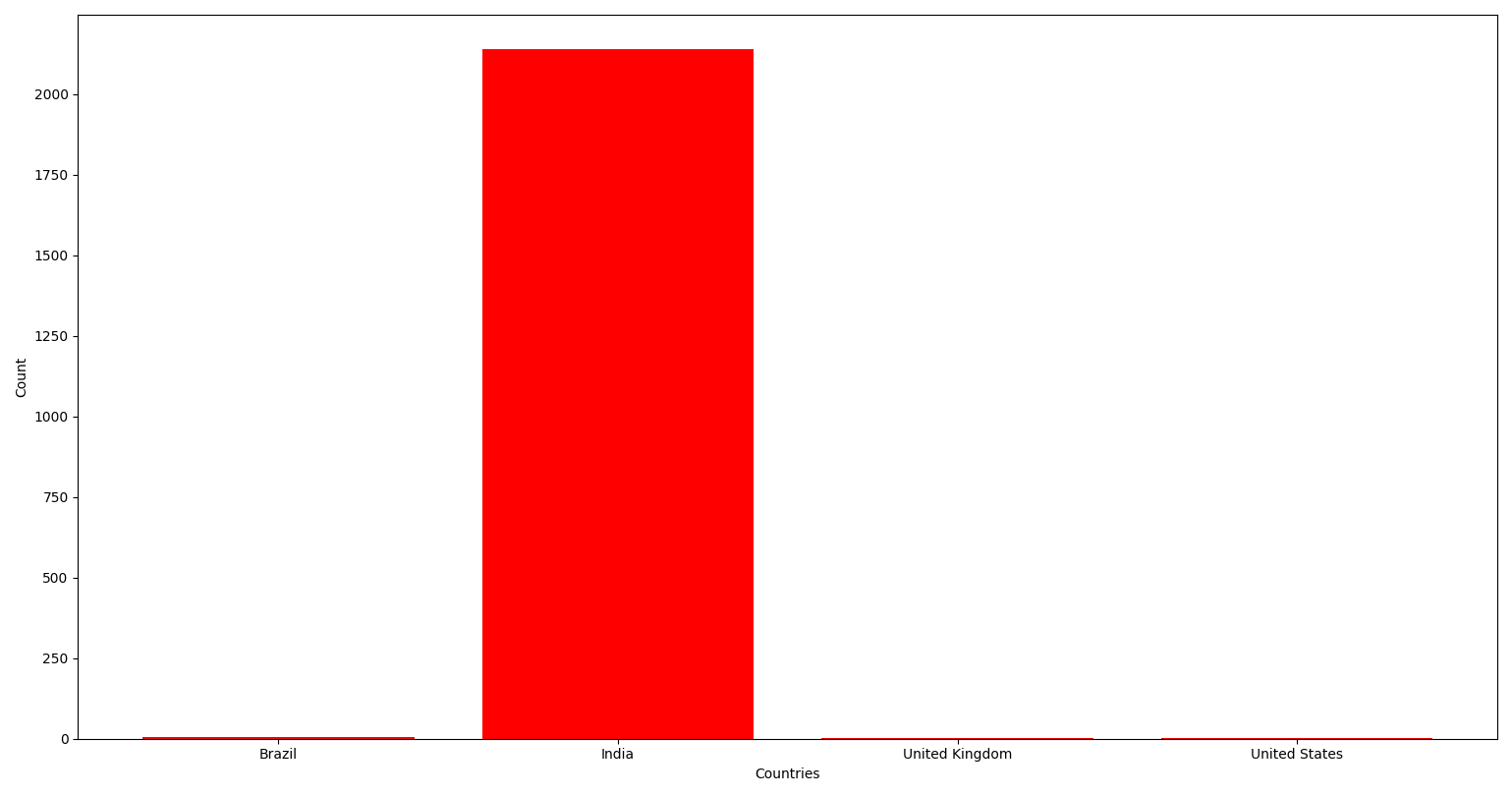
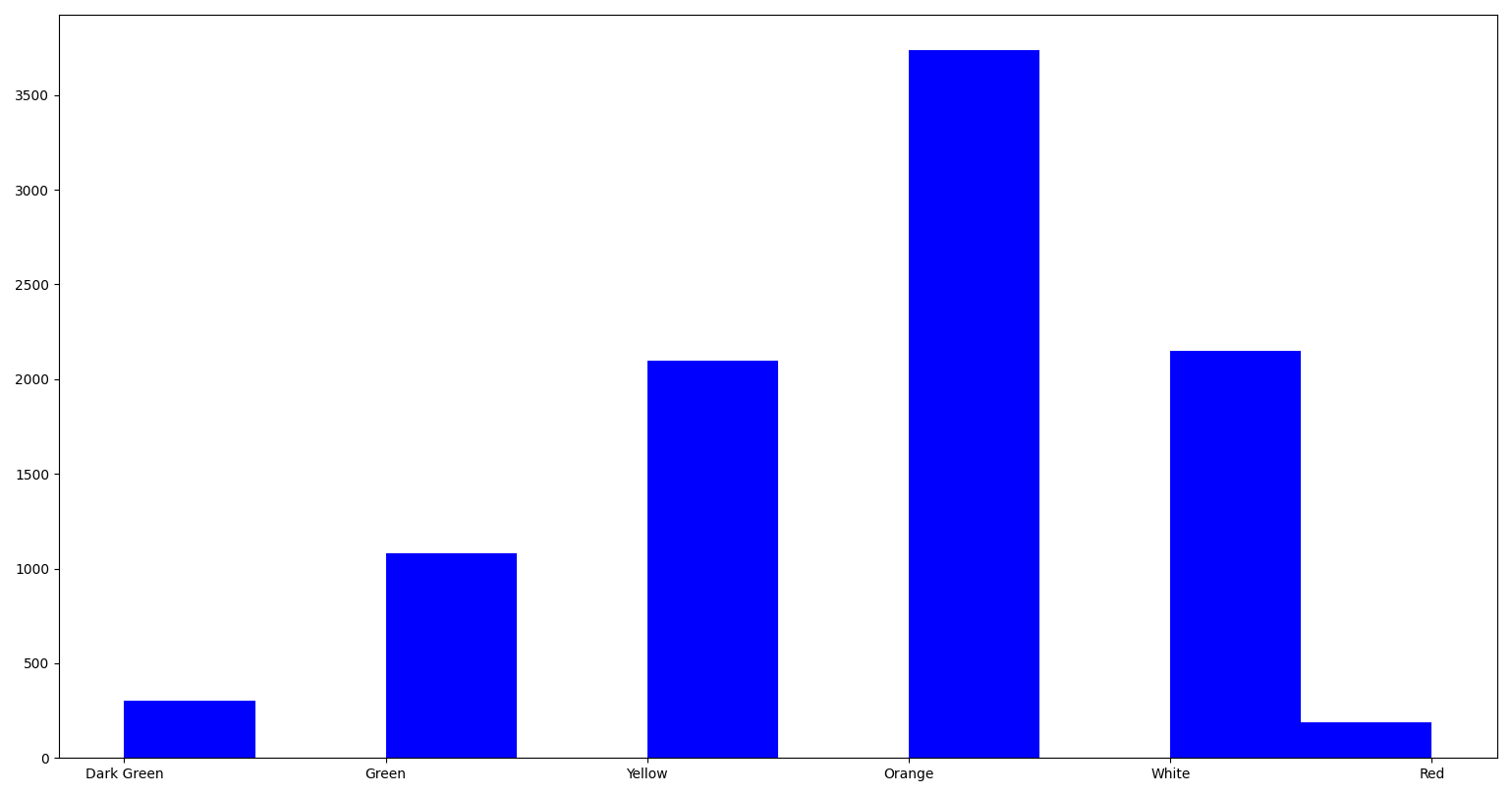
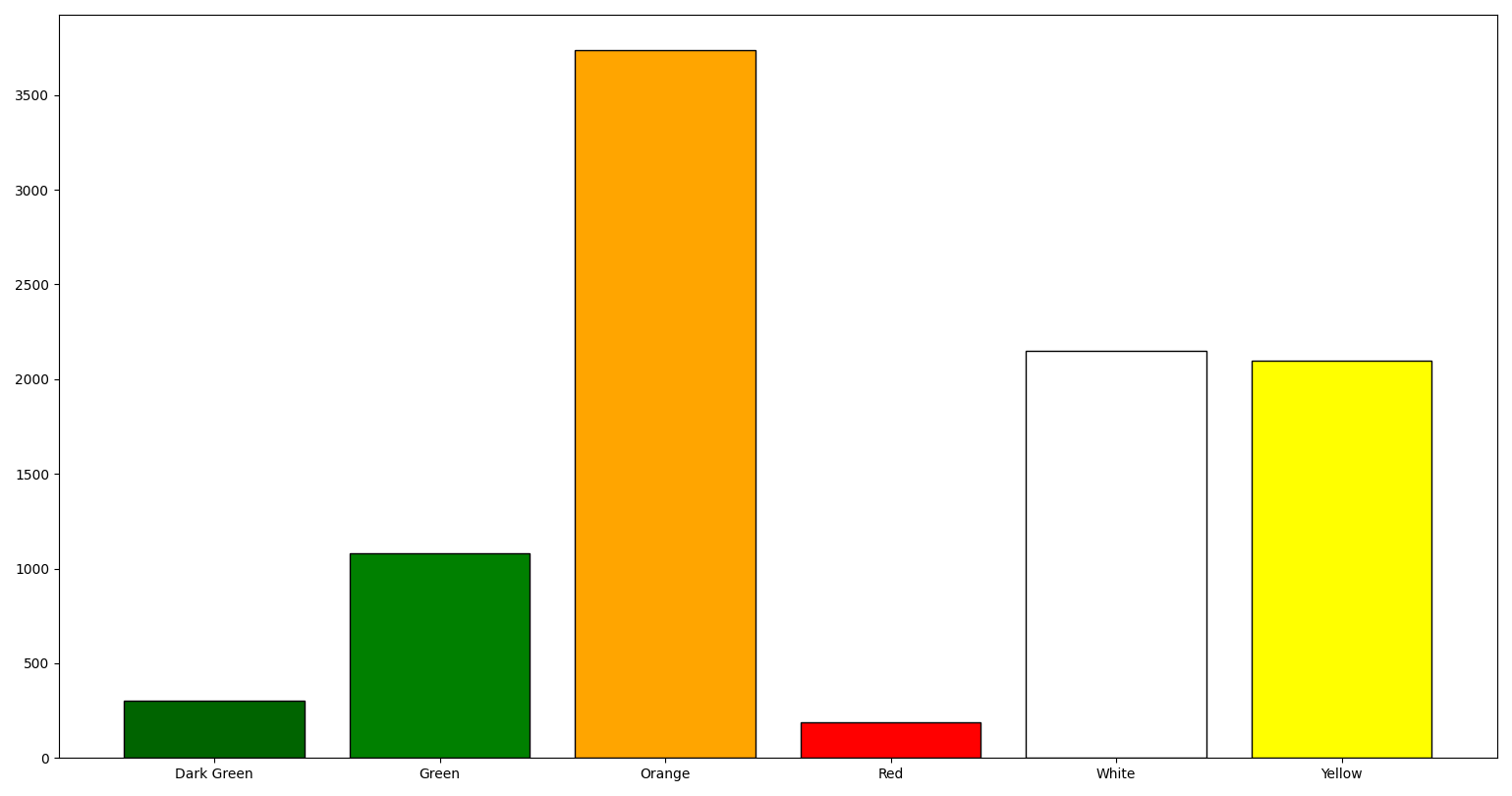
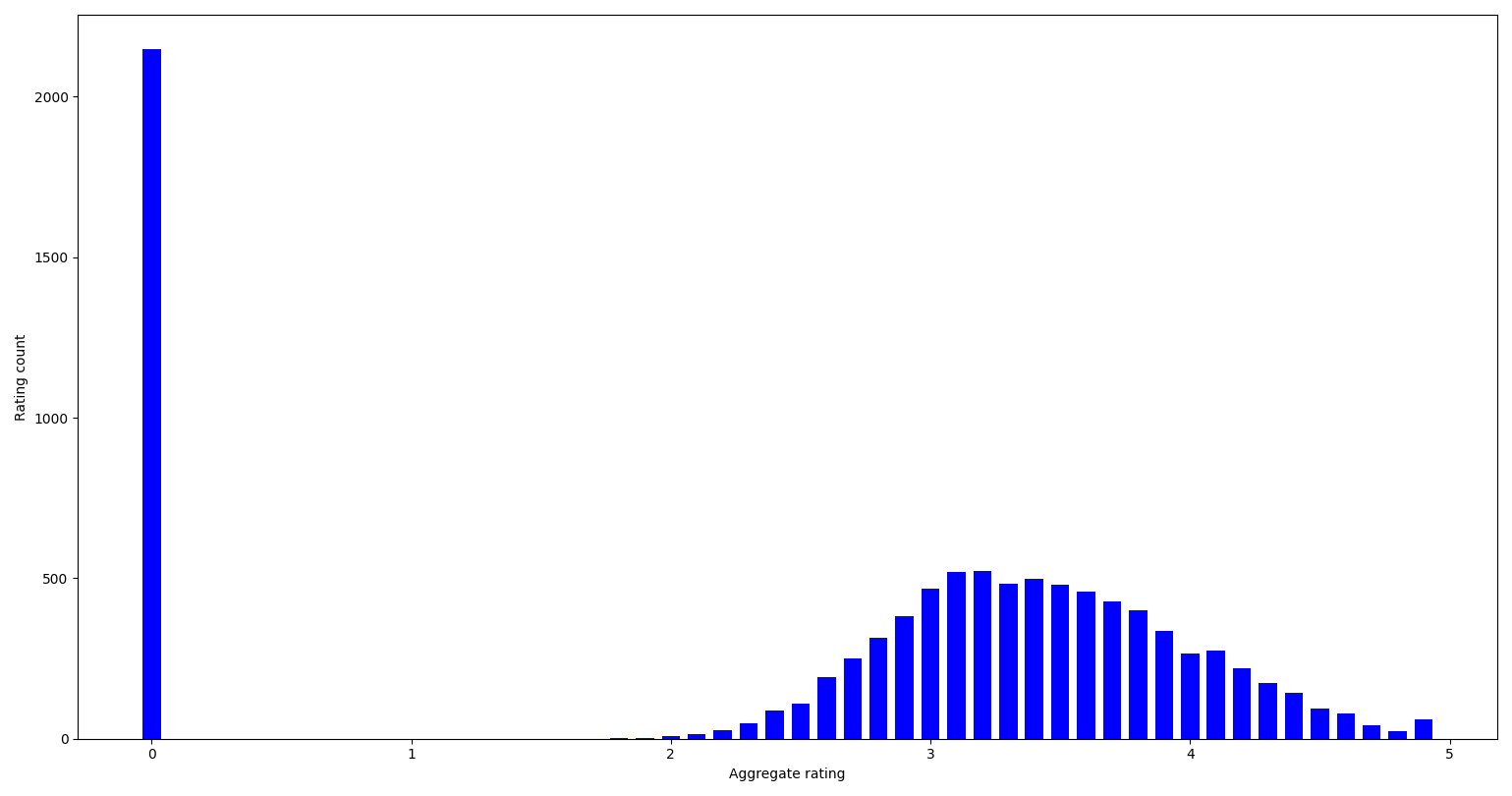
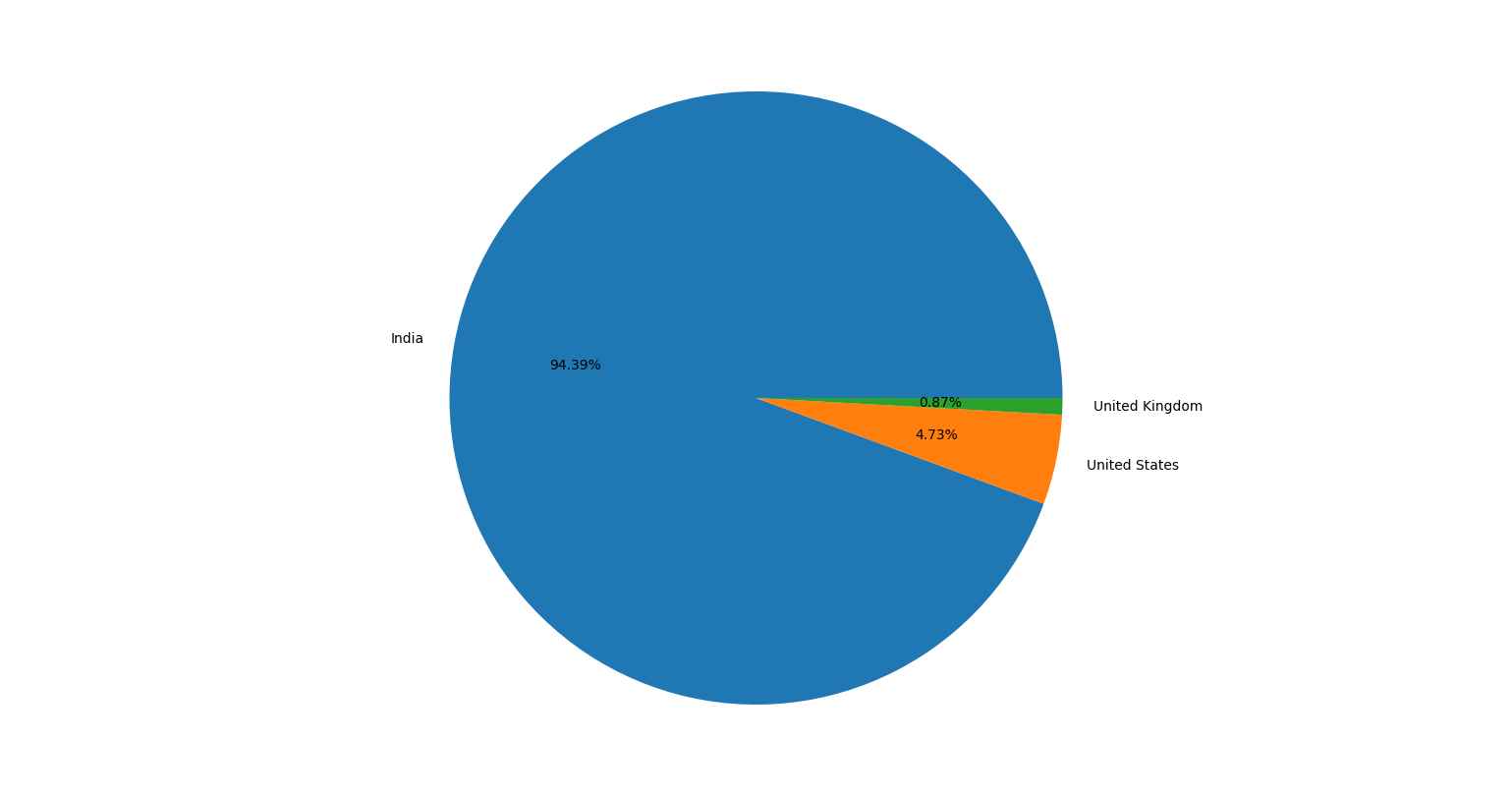
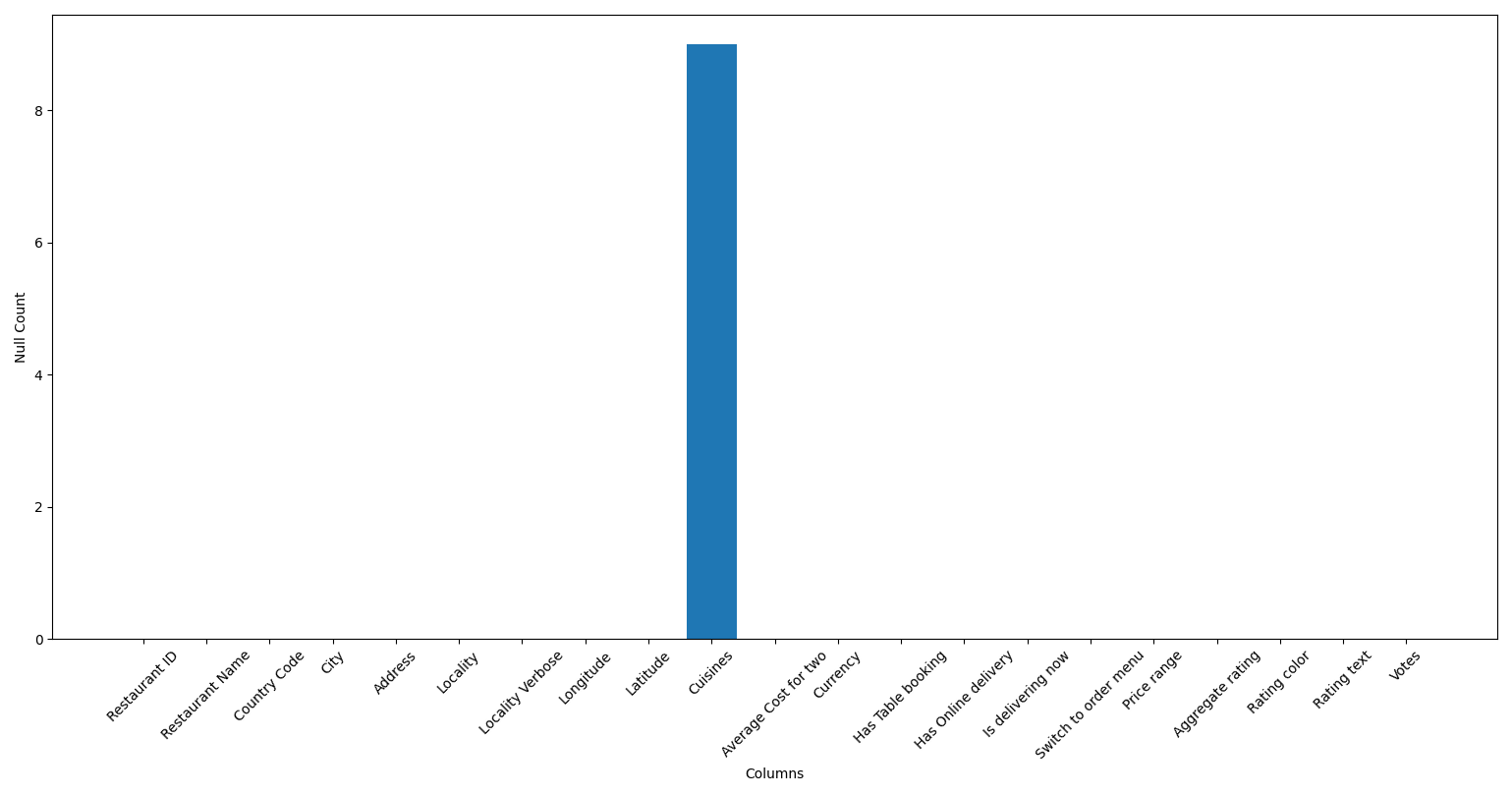
**9274 2800042 Vista - The Park 1 ... Good 74 India**

**9275 2800019 Flying Spaghetti Monster 1 ... Very Good 316 India**

**[8652 rows x 22 columns]**

# SNAPSHOTS:

**Graphs :**

****

# CONCLUSION AND FUTURE SCOPE:

In this project, we utilized the Zomato dataset along with Python libraries such as Pandas, NumPy, and Matplotlib for data analysis. Through this analysis, we gained insights into various aspects of the restaurant industry, including customer preferences, popular cuisines, restaurant ratings, and geographical distribution.

Some of the key findings include:

1.Identification of popular cuisines and their distribution across different regions.

2.Analysis of factors influencing restaurant ratings, such as location, cuisine, and price range.

3.Understanding customer preferences based on reviews and ratings.

4.Exploring trends in restaurant openings and closures over time.

Overall, the analysis provided valuable insights into the dynamics of the restaurant industry, which can be utilized for strategic decision-making and business planning.

Future Scope:

1.Predictive Modeling: Utilize machine learning techniques to predict restaurant ratings or customer preferences based on various features like cuisine type, location, ambience, etc.

2.Sentiment Analysis: Incorporate natural language processing (NLP) to analyze customer reviews and sentiments, providing deeper insights into customer satisfaction and areas for improvement.

3.Recommendation System: Develop a recommendation system to suggest restaurants to users based on their past preferences, location, and other relevant factors.

4.Market Basket Analysis: Explore associations between different food items or categories in orders, enabling targeted marketing strategies or menu optimizations.

5.Real-time Analysis: Implement real-time data processing to capture and analyze dynamic trends in Zomato's restaurant data, allowing for more timely decision-making and interventions.

These future directions can enhance the depth and scope of your data analysis project, providing more actionable insights and value to stakeholders.