eda-vente

September 17, 2024

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

```
[149]: data = pd.read_csv('data/vente.csv', encoding='latin-1')
```

```
[3]: df = data.copy()
     df.head()
```

```
[3]:    Restaurant ID         Restaurant Name  Country Code              City  \
     0        6317637         Le Petit Souffle           162        Makati City
     1        6304287         Izakaya Kikufuji           162        Makati City
     2        6300002   Heat - Edsa Shangri-La           162  Mandaluyong City
     3        6318506                     Ooma           162  Mandaluyong City
     4        6314302              Sambo Kojin           162  Mandaluyong City

                                                  Address  \
     0  Third Floor, Century City Mall, Kalayaan Avenu…
     1  Little Tokyo, 2277 Chino Roces Avenue, Legaspi…
     2  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal…
     3  Third Floor, Mega Fashion Hall, SM Megamall, O…
     4  Third Floor, Mega Atrium, SM Megamall, Ortigas…

                                         Locality  \
     0   Century City Mall, Poblacion, Makati City
     1  Little Tokyo, Legaspi Village, Makati City
     2  Edsa Shangri-La, Ortigas, Mandaluyong City
     3      SM Megamall, Ortigas, Mandaluyong City
     4      SM Megamall, Ortigas, Mandaluyong City

                                       Locality Verbose   Longitude   Latitude  \
     0  Century City Mall, Poblacion, Makati City, Mak…  121.027535  14.565443
     1  Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101  14.553708
     2  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma…  121.056831  14.581404
     3  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.056475  14.585318
```

```
4  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.057508  14.584450

                          Cuisines  …          Currency Has Table booking  \
0       French, Japanese, Desserts  …  Botswana Pula(P)               Yes
1                         Japanese  …  Botswana Pula(P)               Yes
2  Seafood, Asian, Filipino, Indian  …  Botswana Pula(P)               Yes
3                  Japanese, Sushi  …  Botswana Pula(P)                No
4                 Japanese, Korean  …  Botswana Pula(P)               Yes

 Has Online delivery Is delivering now Switch to order menu Price range  \
0                 No                No                   No            3
1                 No                No                   No            3
2                 No                No                   No            4
3                 No                No                   No            4
4                 No                No                   No            4

  Aggregate rating  Rating color Rating text Votes
0              4.8    Dark Green   Excellent   314
1              4.5    Dark Green   Excellent   591
2              4.4         Green   Very Good   270
3              4.9    Dark Green   Excellent   365
4              4.8    Dark Green   Excellent   229

[5 rows x 21 columns]
```

[4]:
```python
df.columns
```

[4]:
```
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')
```

[5]:
```python
df.info()
```

```
<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
```

[6]:
```
df.describe()
```

[6]:
```
       Restaurant ID  Country Code    Longitude     Latitude  \
count   9.551000e+03   9551.000000  9551.000000  9551.000000
mean    9.051128e+06     18.365616    64.126574    25.854381
std     8.791521e+06     56.750546    41.467058    11.007935
min     5.300000e+01      1.000000  -157.948486   -41.330428
25%     3.019625e+05      1.000000    77.081343    28.478713
50%     6.004089e+06      1.000000    77.191964    28.570469
75%     1.835229e+07      1.000000    77.282006    28.642758
max     1.850065e+07    216.000000   174.832089    55.976980

       Average Cost for two  Price range  Aggregate rating         Votes
count           9551.000000  9551.000000       9551.000000   9551.000000
mean            1199.210763     1.804837          2.666370    156.909748
std            16121.183073     0.905609          1.516378    430.169145
min                0.000000     1.000000          0.000000      0.000000
25%              250.000000     1.000000          2.500000      5.000000
50%              400.000000     2.000000          3.200000     31.000000
75%              700.000000     2.000000          3.700000    131.000000
max           800000.000000     4.000000          4.900000  10934.000000
```

[24]:
```
df.shape
```

[24]:
```
(9551, 21)
```

0.1 Que devons nous faire en Analyse de données?

1. Valeurs Manquantes

2. Explorations des variables numériques
3. Exploration des variables catégorielles
4. Trouvez les relations entre les variables

[8]:
```
# Valeurs manquantes
df.isnull().sum()
```

[8]:
```
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
```

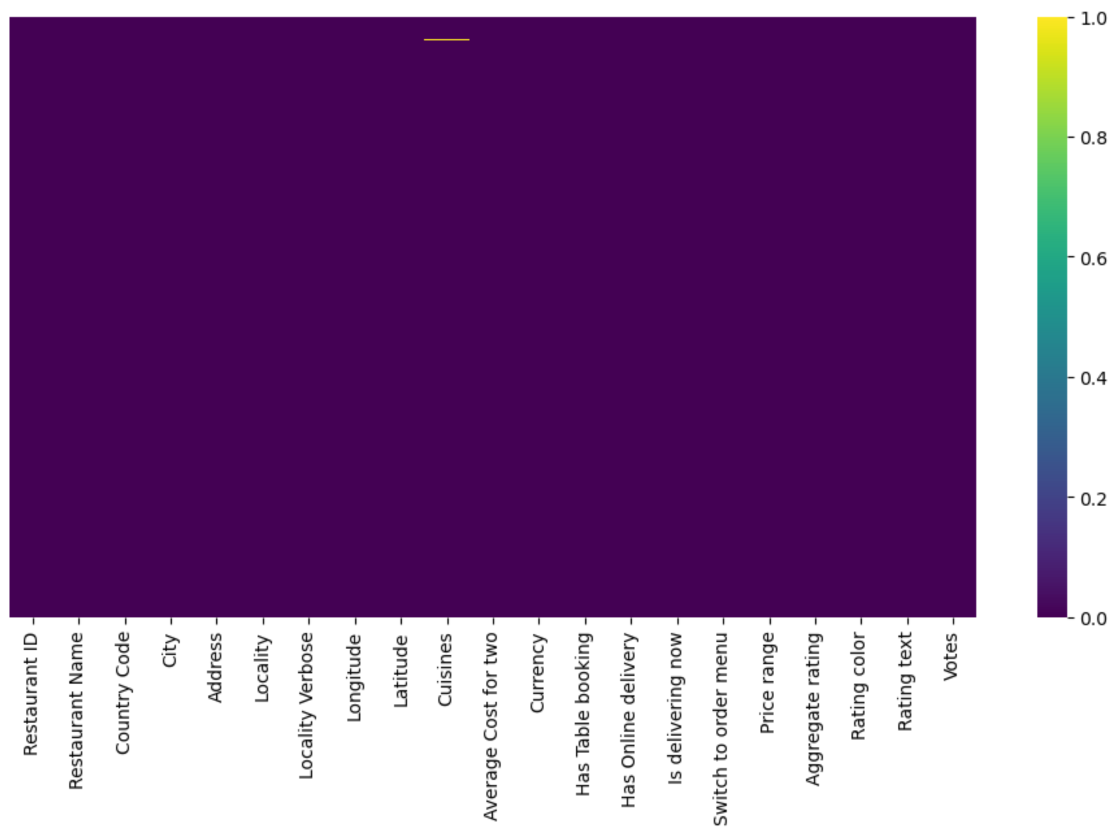
[17]:
```
# valeurs_manquantes = []
# for feature in df.columns:
#     if df[feature].isnull().sum() > 0:
#         valeurs_manquantes.append(feature)

# valeurs_manquantes
# Ou
[feature for feature in df.columns if df[feature].isnull().sum() > 0]
```

[17]:
```
['Cuisines']
```

[73]:
```
# Visualisez avec la carte thermique
plt.rcParams['figure.figsize'] = (12, 6)
sns.heatmap(df.isnull(), yticklabels=False, cmap='viridis')
```

[73]:
```
<Axes: >
```



[27]:
```
df_country = pd.read_excel('data/Country-Code.xlsx')
df_country.head()
```

[27]:
```
   Country Code    Country
0             1      India
1            14  Australia
2            30     Brazil
3            37     Canada
4            94  Indonesia
```

[26]:
```
df.columns
```

[26]:
```
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')
```

[29]:
```python
# Utilisation de merge pour combiner les deux df
final_df = pd.merge(df, df_country, on='Country Code', how='left')
```

[32]:
```python
final_df.head(3)
```

[32]:
```
   Restaurant ID         Restaurant Name  Country Code              City  \
0        6317637        Le Petit Souffle           162       Makati City
1        6304287        Izakaya Kikufuji           162       Makati City
2        6300002  Heat - Edsa Shangri-La           162  Mandaluyong City

                                             Address  \
0  Third Floor, Century City Mall, Kalayaan Avenu…
1  Little Tokyo, 2277 Chino Roces Avenue, Legaspi…
2  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal…

                                    Locality  \
0   Century City Mall, Poblacion, Makati City
1  Little Tokyo, Legaspi Village, Makati City
2  Edsa Shangri-La, Ortigas, Mandaluyong City

                                    Locality Verbose   Longitude   Latitude  \
0  Century City Mall, Poblacion, Makati City, Mak…  121.027535  14.565443
1  Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101  14.553708
2  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma…  121.056831  14.581404

                          Cuisines  …  Has Table booking  \
0       French, Japanese, Desserts  …                Yes
1                         Japanese  …                Yes
2  Seafood, Asian, Filipino, Indian  …                Yes

  Has Online delivery Is delivering now Switch to order menu Price range  \
0                  No                No                   No           3
1                  No                No                   No           3
2                  No                No                   No           4

  Aggregate rating  Rating color  Rating text Votes      Country
0              4.8    Dark Green    Excellent   314  Phillipines
1              4.5    Dark Green    Excellent   591  Phillipines
2              4.4         Green    Very Good   270  Phillipines

[3 rows x 22 columns]
```

[33]:
```python
# V�rifions les types
final_df.dtypes
```

[33]:
```
Restaurant ID              int64
Restaurant Name           object
```

```
Country Code              int64
City                     object
Address                  object
Locality                 object
Locality Verbose         object
Longitude               float64
Latitude                float64
Cuisines                 object
Average Cost for two      int64
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
dtype: object
```

[34]:
```
final_df.columns
```

[34]:
```
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')
```

[35]:
```
final_df['Country'].value_counts()
```

[35]:
```
Country
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: count, dtype: int64
```

[36]:
```python
country_name = final_df['Country'].value_counts().index
country_name
```

[36]:
```
Index(['India', 'United States', 'United Kingdom', 'Brazil', 'UAE',
       'South Africa', 'New Zealand', 'Turkey', 'Australia', 'Phillipines',
       'Indonesia', 'Singapore', 'Qatar', 'Sri Lanka', 'Canada'],
      dtype='object', name='Country')
```

[37]:
```python
country_value = final_df['Country'].value_counts().values
country_value
```

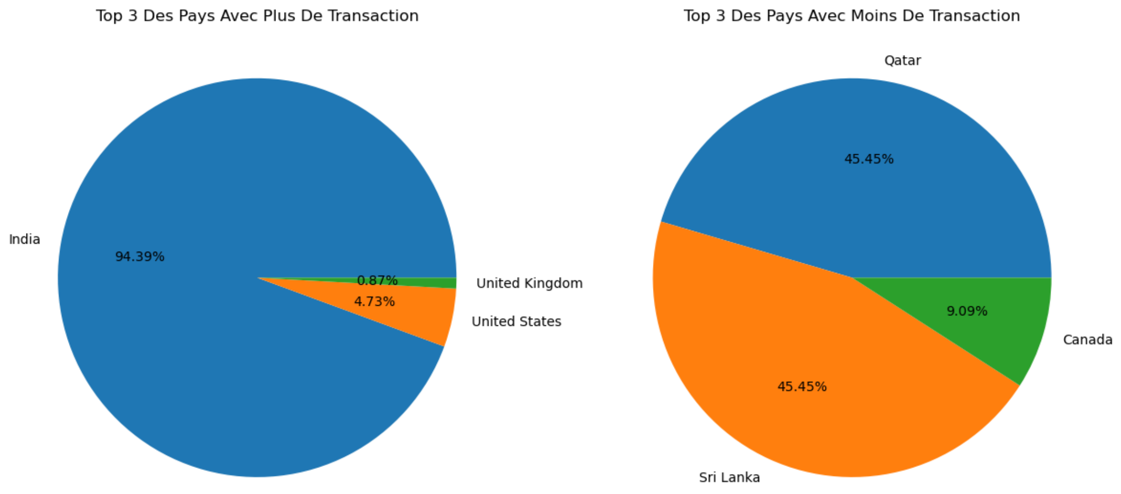
[37]:
```
array([8652,  434,   80,   60,   60,   60,   40,   34,   24,   22,   21,
         20,   20,   20,    4], dtype=int64)
```

[72]:
```python
# Création d'un Pie Chart pour le top 3
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

ax[0].pie(country_value[:3], labels=country_name[:3], autopct='%1.2f%%')
ax[0].set_title('Top 3 Des Pays Avec Plus De Transaction')

ax[1].pie(country_value[-3:], labels=country_name[-3:], autopct='%1.2f%%')
ax[1].set_title('Top 3 Des Pays Avec Moins De Transaction')

plt.tight_layout()
plt.show()
# plt.pie(country_value[:3], labels=country_name[:3], autopct='%1.2f%%')
```



0.2 Observation:

- L'Inde a le plus grand taux de transation effectué suivi respectivement des États-Unis et du Royaume-Uni.
- Le Canada est le pays avec le plus faible taux de transaction suivi du Quatar et du Sri Lanka qui ont le même taux de transactions effectués

```
[52]: final_df.columns
```

```
[52]: 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')
```

```
[60]: ratings = final_df.groupby(['Aggregate rating', 'Rating color', 'Rating text']).
       ↪size().reset_index().rename(columns={0: 'Rating count'})
      ratings
```

```
[60]:     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
```

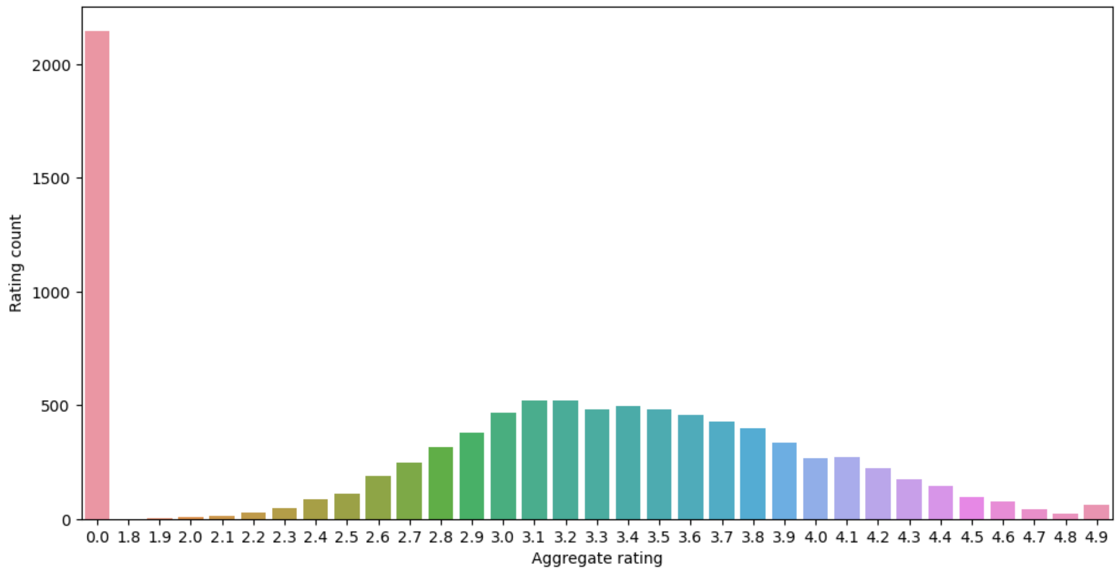
```
23                4.0        Green   Very Good          266
24                4.1        Green   Very Good          274
25                4.2        Green   Very Good          221
26                4.3        Green   Very Good          174
27                4.4        Green   Very Good          144
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
```

0.3 Observation

1. Quand la note est comprise entre 4.5 et 4.9 (4.5 <= Note <= 4.9) ————> Excellent
2. Quand la note est comprise entre 4.1 et 4.4 (4.1 <= Note <= 4.4) ————> Très bien
3. Quand la note est comprise entre 3.5 et 3.9 (3.5 <= Note <= 3.9) ————> Bien
4. Quand la note est comprise entre 2.5 et 3.4 (2.5 <= Note <= 3.4) ————> Moyen
5. Quand la note est comprise entre 1.8 et 2.4 (1.8 <= Note <= 2.4) ————> Mauvais

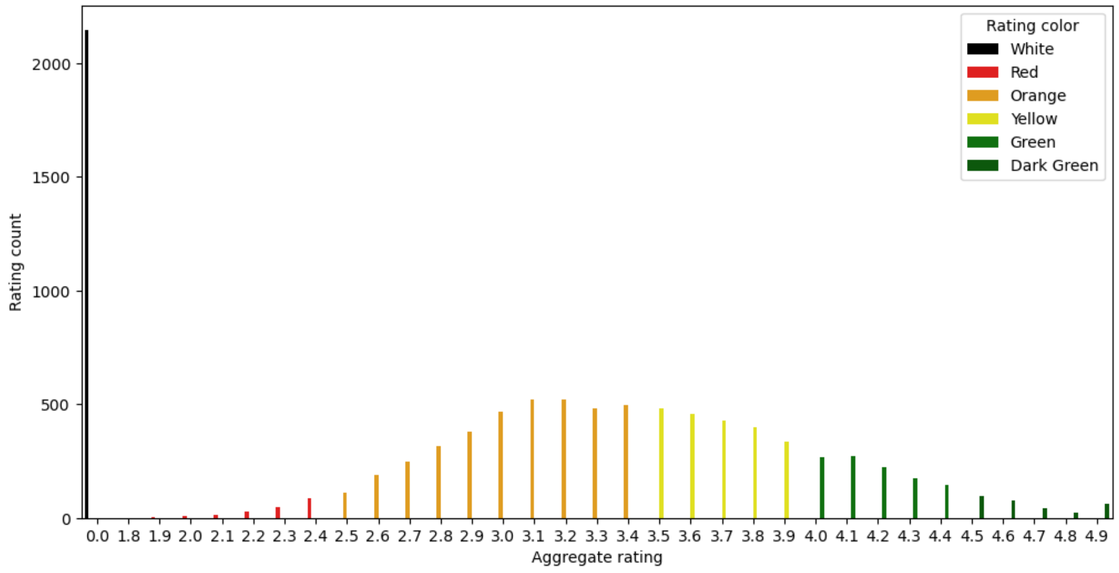
[74]:
```
sns.barplot(x='Aggregate rating', y='Rating count', data=ratings)
```

[74]:
```
<Axes: xlabel='Aggregate rating', ylabel='Rating count'>
```



[82]:
```
sns.barplot(x='Aggregate rating', y='Rating count', hue='Rating color',␣
↪data=ratings, palette=['black', 'red', 'orange', 'yellow', 'green',␣
↪'darkgreen'])
```

[82]:
```
<Axes: xlabel='Aggregate rating', ylabel='Rating count'>
```

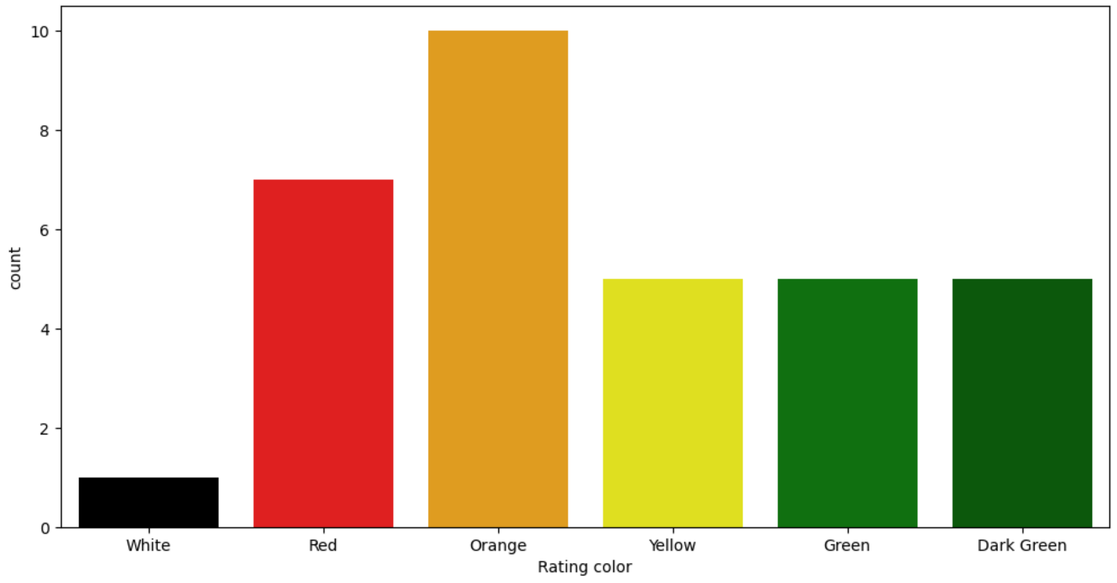


0.4 Observation

1. Les personnes qui n'ont pas votées sont les plus
2. Les notes les plus élévées se trouves en 2.9 et 3.8 (2.9 <= Note <= 3.8)

[84]:
```python
sns.countplot(x='Rating color', data=ratings, palette=['black', 'red',␣
↪'orange', 'yellow', 'green', 'darkgreen'])
```

[84]:
```
<Axes: xlabel='Rating color', ylabel='count'>
```



[85]:
```
ratings
```

[85]:
```
    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
```

[109]:
```
# Les pays qui n'ont pas attribués de note
country_not_ratings = final_df[final_df['Rating text'] == 'Not rated'].
 ↪groupby(['Country']).size().reset_index().rename(columns={0: 'Country␣
 ↪count'})
```

[110]:
```
country_not_ratings
```

[110]:
```
         Country  Country count
0         Brazil              5
1          India           2139
```

```
2  United Kingdom              1
3   United States              3
```

```
[121]: country_poor_ratings = final_df[final_df['Rating text'] == 'Poor'].
         ↪groupby(['Country']).size().reset_index().rename(columns={0: 'Country␣
         ↪count'})
```

```
[122]: country_poor_ratings
```

```
[122]:          Country  Country count
       0      Australia              1
       1          India            180
       2    New Zealand              1
       3      Sri Lanka              1
       4            UAE              1
       5  United States              2
```

```
[118]: country_Excellent_ratings = final_df[final_df['Rating text'] == 'Excellent'].
         ↪groupby(['Country']).size().reset_index().rename(columns={0: 'Country␣
         ↪count'})
```

```
[119]: country_Excellent_ratings
```

```
[119]:            Country  Country count
       0        Australia              1
       1           Brazil             16
       2            India            116
       3        Indonesia              7
       4      New Zealand             12
       5      Phillipines             12
       6            Qatar              4
       7     South Africa             12
       8        Sri Lanka              2
       9           Turkey             10
       10             UAE             18
       11  United Kingdom             23
       12   United States             68
```

0.5 Observation

- L'inde est le pays le mieux noté, le plus mal noté et celui avec le plus de transaction mal noté
- On constate aussi qu'en terme rappport nombre de transaction et de qualité les États-Unis ont un excellent services en se basant sur les notes Excellente et mauvaise

```
[126]: # Trouver les dévises utilisées par les pays
       final_df.columns
```

```
[126]: 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')
```

```
[136]: final_df.groupby(['Country', 'Currency']).size().reset_index().
        ↪rename(columns={0: 'Count currency'})
```

```
[136]:            Country                Currency  Count currency
       0         Australia              Dollar($)              24
       1            Brazil      Brazilian Real(R$)              60
       2            Canada               Dollar($)               4
       3             India        Indian Rupees(Rs.)          8652
       4         Indonesia  Indonesian Rupiah(IDR)              21
       5       New Zealand          NewZealand($)              40
       6       Phillipines        Botswana Pula(P)              22
       7             Qatar          Qatari Rial(QR)              20
       8         Singapore               Dollar($)              20
       9      South Africa                 Rand(R)              60
       10        Sri Lanka   Sri Lankan Rupee(LKR)              20
       11           Turkey        Turkish Lira(TL)              34
       12              UAE       Emirati Diram(AED)              60
       13   United Kingdom              Pounds(£)              80
       14    United States               Dollar($)             434
```

```
[137]: # Vérifiez les pays qui utilisent le service de livraison en ligne
       final_df.groupby(['Country', 'Has Online delivery']).size().reset_index().
        ↪rename(columns={0: 'Count Online delivery'})
```

```
[137]:            Country Has Online delivery  Count Online delivery
       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
```

0.6 Observation

- Les dévises les plus utilisées sont respectivement le Indian Rupees(Rs.) et le Dollar($) avec le Indian Rupees(Rs.) en très grande majorité
- Seul l'Inde et l'UAE utilise le service de livraison en ligne

[ ]:

```

```

[143]:

```
# Top 5 des villes qui ont effectuées plus de transactions
city_values = final_df.City.value_counts().values
city_labels = final_df.City.value_counts().index
print(city_values)
print(city_labels)
```

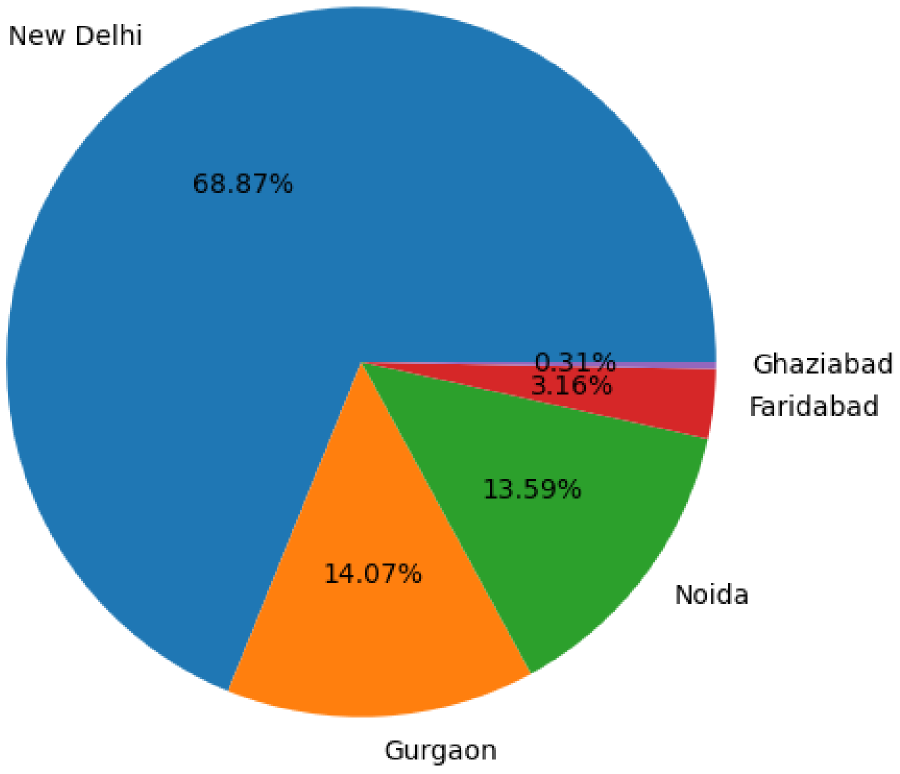
```
[5473 1118 1080  251   25   21   21   21   21   21   20   20   20   20
   20   20   20   20   20   20   20   20   20   20   20   20   20   20
   20   20   20   20   20   20   20   20   20   20   20   20   20   20
   20   20   20   20   20   20   20   20   20   20   20   20   20   20
   20   20   20   20   20   20   20   20   20   20   20   20   20   20
   20   20   20   20   20   20   20   18   18   16   14   11    6    4
    4    3    3    2    2    2    2    2    2    2    2    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1    1    1    1    1    1    1    1    1    1    1    1    1    1
    1]
Index(['New Delhi', 'Gurgaon', 'Noida', 'Faridabad', 'Ghaziabad',
       'Bhubaneshwar', 'Amritsar', 'Ahmedabad', 'Lucknow', 'Guwahati',
       …
       'Ojo Caliente', 'Montville', 'Monroe', 'Miller', 'Middleton Beach',
       'Panchkula', 'Mc Millan', 'Mayfield', 'Macedon', 'Vineland Station'],
      dtype='object', name='City', length=141)
```

[145]:

```
plt.pie(city_values[:5], labels=city_labels[:5], autopct='%1.2f%%')
```

[145]:

```
([<matplotlib.patches.Wedge at 0x286e6717c10>,
  <matplotlib.patches.Wedge at 0x286e6715090>,
  <matplotlib.patches.Wedge at 0x286e6716190>,
  <matplotlib.patches.Wedge at 0x286e6714510>,
  <matplotlib.patches.Wedge at 0x286e4a77790>],
 [Text(-0.6145352824185932, 0.9123301960708633, 'New Delhi'),
  Text(0.0623675251198054, -1.0982305276263407, 'Gurgaon'),
  Text(0.8789045225625368, -0.6614581167535246, 'Noida'),
  Text(1.0922218418223437, -0.13058119407559224, 'Faridabad'),
  Text(1.099946280005612, -0.010871113182029924, 'Ghaziabad')],
 [Text(-0.3352010631374145, 0.497634652402289, '68.87%'),
  Text(0.0340186500653484, -0.5990348332507311, '14.07%'),
  Text(0.47940246685229276, -0.36079533641101336, '13.59%'),
  Text(0.5957573682667329, -0.07122610585941394, '3.16%'),
  Text(0.5999706981848791, -0.005929698099289049, '0.31%')])
```



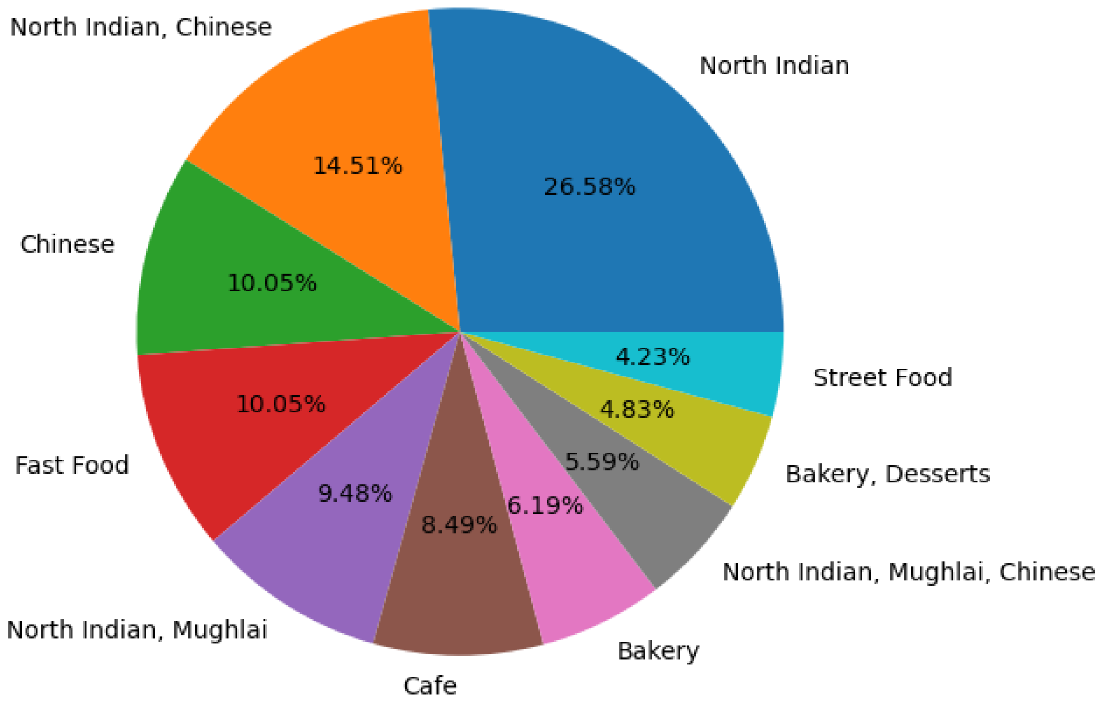
```
[146]: # Top 10 des cuisines
       final_df.columns
```

```
[146]: 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')
```

```
[148]: cuisine_values = final_df.Cuisines.value_counts().values
       cuisine_labels = final_df.Cuisines.value_counts().index

       plt.pie(cuisine_values[:10], labels=cuisine_labels[:10], autopct='%1.2f%%')
```

```
[148]: ([<matplotlib.patches.Wedge at 0x286e4bbc650>,
          <matplotlib.patches.Wedge at 0x286e65e6f10>,
          <matplotlib.patches.Wedge at 0x286e37dc250>,
          <matplotlib.patches.Wedge at 0x286e37dee90>,
          <matplotlib.patches.Wedge at 0x286e85f0310>,
          <matplotlib.patches.Wedge at 0x286e81e9ed0>,
          <matplotlib.patches.Wedge at 0x286d4ed5bd0>,
          <matplotlib.patches.Wedge at 0x286e7834d10>,
          <matplotlib.patches.Wedge at 0x286e676d5d0>,
          <matplotlib.patches.Wedge at 0x286e81e9b50>],
         [Text(0.7383739846958008, 0.8153550507137645, 'North Indian'),
          Text(-0.5794679314239953, 0.9349956772366362, 'North Indian, Chinese'),
          Text(-1.067309479615702, 0.26617752482593154, 'Chinese'),
          Text(-1.0185984499802057, -0.4152796620326146, 'Fast Food'),
          Text(-0.5935788454809928, -0.9261015895664211, 'North Indian, Mughlai'),
          Text(-0.005887079599915552, -1.0999842463843672, 'Cafe'),
          Text(0.4842062514572988, -0.9876964645323336, 'Bakery'),
          Text(0.808736477166136, -0.7456174022251013, 'North Indian, Mughlai,
        Chinese'),
          Text(1.0055375294202338, -0.44597564611473206, 'Bakery, Desserts'),
          Text(1.090298995560443, -0.14576728123927227, 'Street Food')],
         [Text(0.4027494461977095, 0.4447391185711442, '26.58%'),
          Text(-0.316073417140361, 0.5099976421290743, '14.51%'),
          Text(-0.5821688070631101, 0.14518774081414446, '10.05%'),
          Text(-0.5555991545346576, -0.22651617929051704, '10.05%'),
          Text(-0.32377027935326874, -0.5051463215816842, '9.48%'),
          Text(-0.003211134327226664, -0.5999914071187457, '8.49%'),
          Text(0.26411250079489024, -0.5387435261085456, '6.19%'),
          Text(0.441128987545165, -0.40670040121369155, '5.59%'),
          Text(0.5484750160474001, -0.24325944333530836, '4.83%'),
          Text(0.5947085430329688, -0.07950942613051214, '4.23%')])
```



0.7 Observation

- La cuisine North Indian est la plus consommée tandisque la cuisine Street Food est la moins consommée

```
[ ]:
```