zomato-data-analysis

August 26, 2024

1 Zomato Data Analysis

2 Step1:Importing the Libaries

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

3 Step 2: Import dataset and Read Dataset

[2]:
```
Dataframe=pd.read_csv(r"C:\Preet\Zomato data .csv")
```

[3]:
```
Dataframe.head()
```

[3]:
```
                  name online_order book_table   rate  votes  \
0                Jalsa          Yes        Yes  4.1/5    775
1        Spice Elephant          Yes         No  4.1/5    787
2       San Churro Cafe          Yes         No  3.8/5    918
3  Addhuri Udupi Bhojana           No         No  3.7/5     88
4         Grand Village            No         No  3.8/5    166

   approx_cost(for two people) listed_in(type)
0                          800          Buffet
1                          800          Buffet
2                          800          Buffet
3                          300          Buffet
4                          600          Buffet
```

[4]:
```
Dataframe
```

[4]:
```
                    name online_order book_table   rate  votes  \
0                  Jalsa          Yes        Yes  4.1/5    775
1          Spice Elephant          Yes         No  4.1/5    787
2         San Churro Cafe          Yes         No  3.8/5    918
3   Addhuri Udupi Bhojana           No         No  3.7/5     88
```

```
4            Grand Village            No          No  3.8/5    166
..                     …           …         …    …    …
143       Melting Melodies            No          No  3.3/5      0
144        New Indraprasta            No          No  3.3/5      0
145           Anna Kuteera           Yes          No  4.0/5    771
146                 Darbar            No          No  3.0/5     98
147          Vijayalakshmi           Yes          No  3.9/5     47

     approx_cost(for two people) listed_in(type)
0                            800          Buffet
1                            800          Buffet
2                            800          Buffet
3                            300          Buffet
4                            600          Buffet
..                           …             …
143                          100          Dining
144                          150          Dining
145                          450          Dining
146                          800          Dining
147                          200          Dining

[148 rows x 7 columns]
```

4 Convert datatype Column –Rate

[5]:
```
def HandleRate(value):
    value=str(value).split('/')
    value=value[0];
    return float(value)

Dataframe['rate']=Dataframe['rate'].apply(HandleRate)
print(Dataframe.head())
```

```
                  name online_order book_table  rate  votes  \
0                Jalsa          Yes        Yes   4.1    775
1        Spice Elephant          Yes         No   4.1    787
2       San Churro Cafe          Yes         No   3.8    918
3  Addhuri Udupi Bhojana           No         No   3.7     88
4         Grand Village            No         No   3.8    166

   approx_cost(for two people) listed_in(type)
0                          800          Buffet
1                          800          Buffet
2                          800          Buffet
3                          300          Buffet
4                          600          Buffet
```

5 Summary of dataset

[6]:
```
Dataframe.info()
```

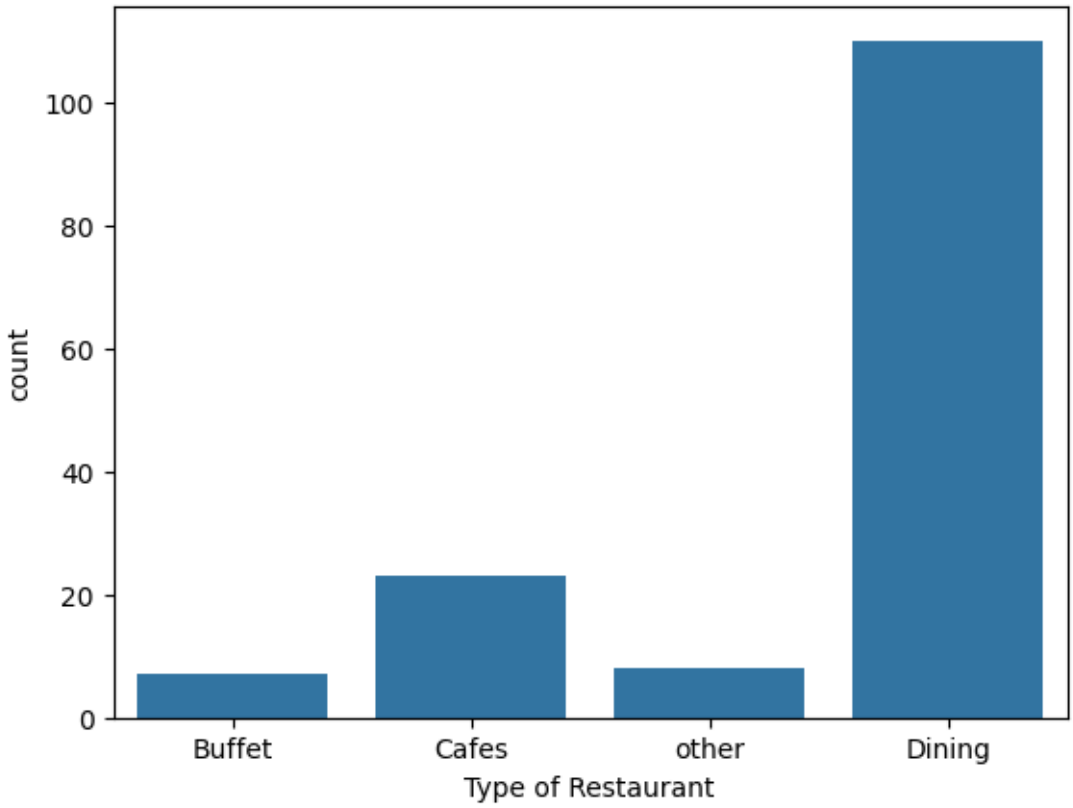
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   name                         148 non-null    object
 1   online_order                 148 non-null    object
 2   book_table                   148 non-null    object
 3   rate                         148 non-null    float64
 4   votes                        148 non-null    int64
 5   approx_cost(for two people)  148 non-null    int64
 6   listed_in(type)              148 non-null    object
dtypes: float64(1), int64(2), object(4)
memory usage: 8.2+ KB
```

6 Conculsion- There is no null values in dataset

7 Type of Restaurant

[7]:
```
sns.countplot(x=Dataframe['listed_in(type)'])
plt.xlabel("Type of Restaurant")
```

[7]:
```
Text(0.5, 0, 'Type of Restaurant')
```



8 Conclusion: The Majority of Restaurants fall into the category in Dining

9 A Large Number Of Individual prefers Dining Restaurants

[8]:
```
Dataframe.head()
```

[8]:

	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
0	Jalsa	Yes	Yes	4.1	775	800	Buffet
1	Spice Elephant	Yes	No	4.1	787	800	Buffet
2	San Churro Cafe	Yes	No	3.8	918	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	300	Buffet
4	Grand Village	No	No	3.8	166		

```
4                          600          Buffet
```

[ ]:

[14]:
```
import pandas as pd
my_str = "Hello!"  # Rename 'pd' to something else
df = pd.DataFrame({'votes': grouped_data})
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[14], line 3
      1 import pandas as pd
      2 my_str = "Hello!"  # Rename 'pd' to something else
----> 3 df = pd.DataFrame({'votes': grouped_data})

NameError: name 'grouped_data' is not defined
```

[15]:
```
import pandas as pd
df = pd.DataFrame({'votes': grouped_data})
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[15], line 2
      1 import pandas as pd
----> 2 df = pd.DataFrame({'votes': grouped_data})

NameError: name 'grouped_data' is not defined
```

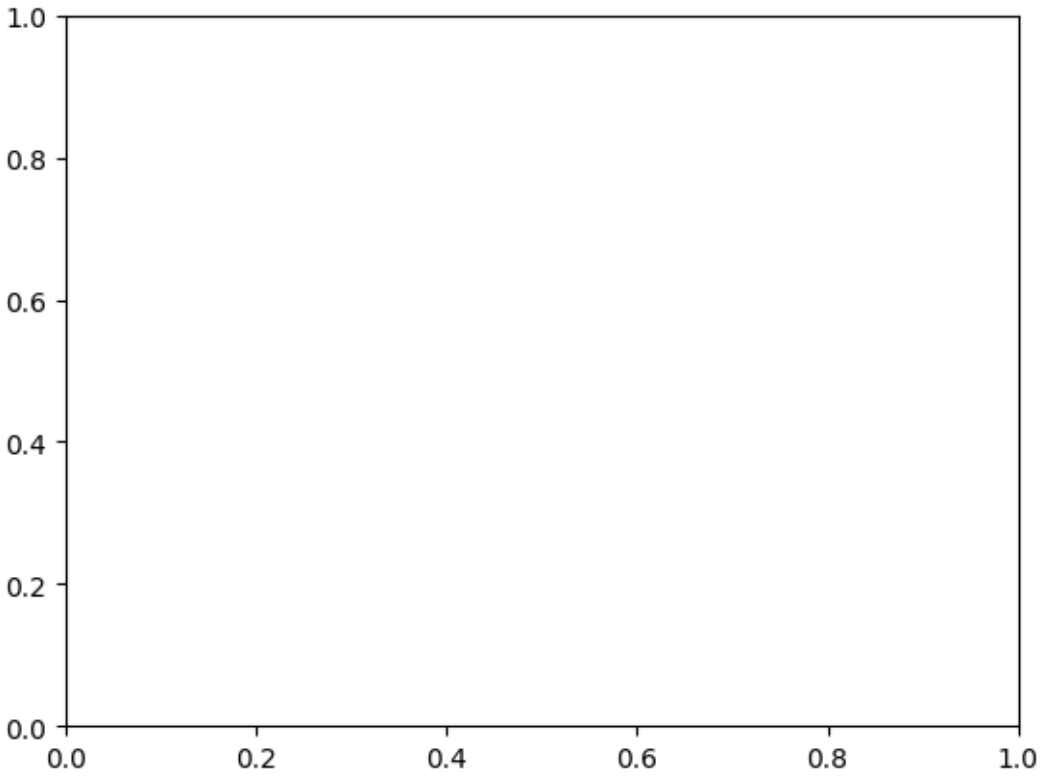
[18]:
```
grouped_data =Dataframe.groupby('listed_in(type)')['votes'].sum()
```

[19]:
```
plt.plot(Dataframe,c="green",marker="o")
plt.xlabel("Type of Restaurnts",c="red",size=20)
plt.ylabel("Votes",c="red",size=20)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[19], line 1
----> 1 plt.plot(Dataframe,c="green",marker="o")
      2 plt.xlabel("Type of Restaurnts",c="red",size=20)
      3 plt.ylabel("Votes",c="red",size=20)

File ~\python1\envs\notebook-7.0.8\Lib\site-packages\matplotlib\pyplot.py:3708,
 ↪in plot(scalex, scaley, data, *args, **kwargs)
   3700 @_copy_docstring_and_deprecators(Axes.plot)
   3701 def plot(
```

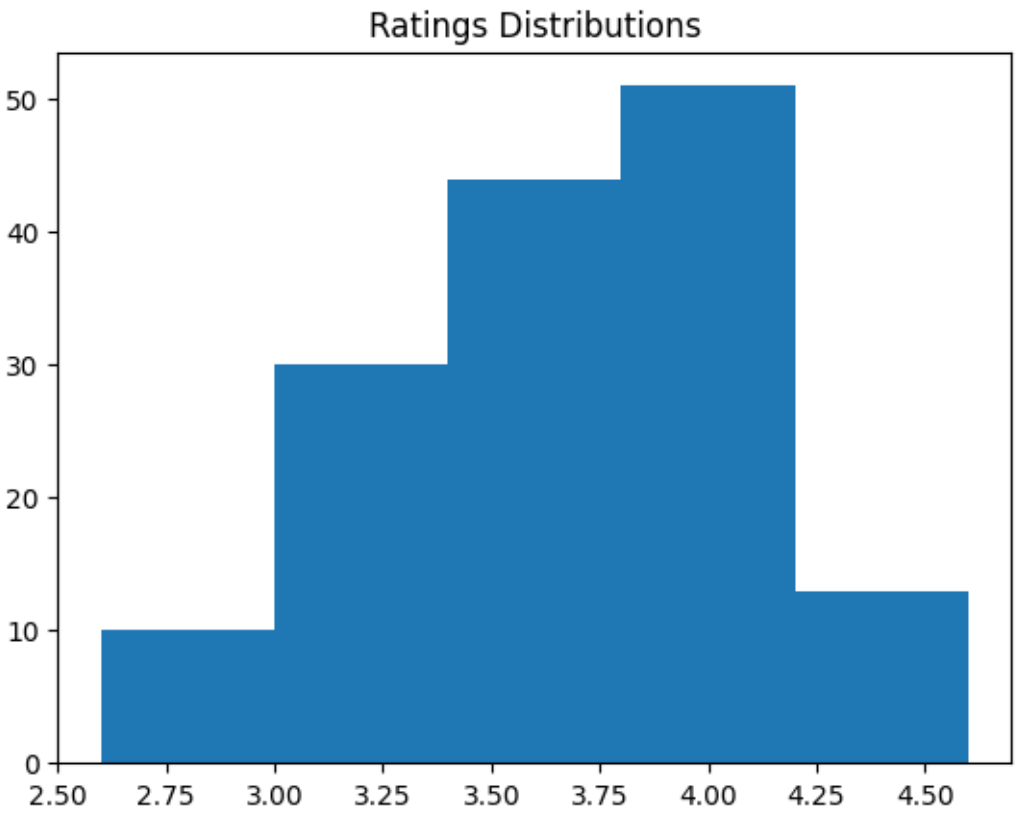
```
   3702     *args: float | ArrayLike | str,
   (…)
   3706     **kwargs,
   3707 ) -> list[Line2D]:
-> 3708     return gca().plot(
   3709         *args,
   3710         scalex=scalex,
   3711         scaley=scaley,
   3712         **({"data": data} if data is not None else {}),
   3713         **kwargs,
   3714     )

File ~\python1\envs\notebook-7.0.8\Lib\site-packages\matplotlib\axes\_axes.py:
 ↪1779, in Axes.plot(self, scalex, scaley, data, *args, **kwargs)
   1536 """
   1537 Plot y versus x as lines and/or markers.
   1538
   (…)
   1776 (``'green'``) or hex strings (``'#008000'``).
   1777 """
   1778 kwargs = cbook.normalize_kwargs(kwargs, mlines.Line2D)
-> 1779 lines = [*self._get_lines(self, *args, data=data, **kwargs)]
   1780 for line in lines:
   1781     self.add_line(line)

File ~\python1\envs\notebook-7.0.8\Lib\site-packages\matplotlib\axes\_base.py:
 ↪296, in _process_plot_var_args.__call__(self, axes, data, *args, **kwargs)
    294     this += args[0],
    295     args = args[1:]
--> 296 yield from self._plot_args(
    297     axes, this, kwargs, ambiguous_fmt_datakey=ambiguous_fmt_datakey)

File ~\python1\envs\notebook-7.0.8\Lib\site-packages\matplotlib\axes\_base.py:
 ↪483, in _process_plot_var_args._plot_args(self, axes, tup, kwargs,␣
 ↪return_kwargs, ambiguous_fmt_datakey)
    481     axes.xaxis.update_units(x)
    482 if axes.yaxis is not None:
--> 483     axes.yaxis.update_units(y)
    485 if x.shape[0] != y.shape[0]:
    486     raise ValueError(f"x and y must have same first dimension, but "
    487                      f"have shapes {x.shape} and {y.shape}")

File ~\python1\envs\notebook-7.0.8\Lib\site-packages\matplotlib\axis.py:1747, i␣
 ↪Axis.update_units(self, data)
   1745 neednew = self.converter != converter
   1746 self.converter = converter
-> 1747 default = self.converter.default_units(data, self)
   1748 if default is not None and self.units is None:
```

```
   1749     self.set_units(default)

File ~\python1\envs\notebook-7.0.8\Lib\site-packages\matplotlib\category.py:105
 ↪in StrCategoryConverter.default_units(data, axis)
    103 # the conversion call stack is default_units -> axis_info -> convert
    104 if axis.units is None:
--> 105     axis.set_units(UnitData(data))
    106 else:
    107     axis.units.update(data)

File ~\python1\envs\notebook-7.0.8\Lib\site-packages\matplotlib\category.py:181
 ↪in UnitData.__init__(self, data)
    179 self._counter = itertools.count()
    180 if data is not None:
--> 181     self.update(data)

File ~\python1\envs\notebook-7.0.8\Lib\site-packages\matplotlib\category.py:214
 ↪in UnitData.update(self, data)
    212 # check if convertible to number:
    213 convertible = True
--> 214 for val in OrderedDict.fromkeys(data):
    215     # OrderedDict just iterates over unique values in data.
    216     _api.check_isinstance((str, bytes), value=val)
    217     if convertible:
    218         # this will only be called so long as convertible is True.

TypeError: unhashable type: 'numpy.ndarray'
```



10 The majority received ratings

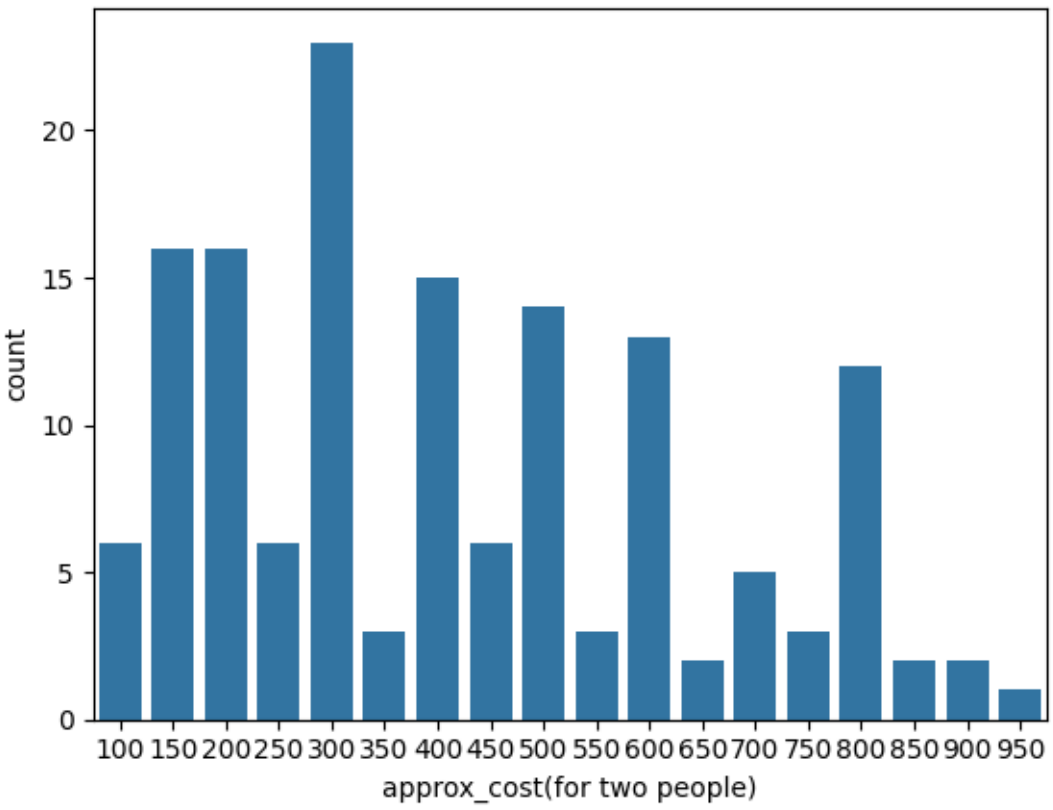
[10]:
```
plt.hist(Dataframe['rate'],bins=5)
plt.title("Ratings Distributions")
plt.show()
```



11 Conclusion: The majority of restaurants received ratings ranging from 3.5 to 4

[11]:
```
couple_data=Dataframe['approx_cost(for two people)']
sns.countplot(x=couple_data)
```

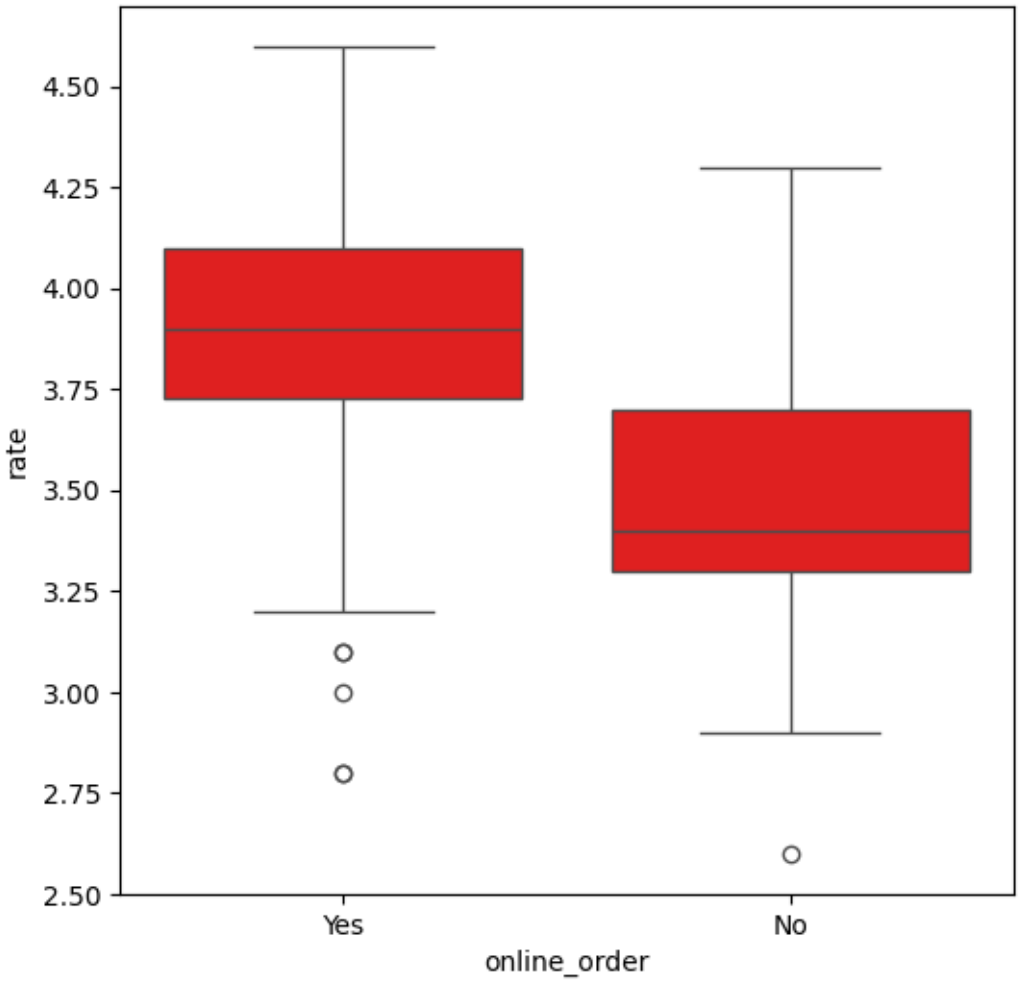
[11]:
```
<Axes: xlabel='approx_cost(for two people)', ylabel='count'>
```



12 Whether online orders receive higher ratings than offline orders

[13]:
```
plt.figure(figsize=(6,6))
sns.boxplot(x='online_order',y='rate',color='red',data=Dataframe)
```

[13]:
```
<Axes: xlabel='online_order', ylabel='rate'>
```



13 Conclusion: Offline orders received lower ratings in comparison to online orders, which obtained excellent ratings

[ ]:
```
pivot_table = Dataframe.pivot_table(index='')
```