ZOMATO DATASET EXERCISE ASSIGNMENT - SUBMITTED BY ABDHAHEER.A

1.IMPORTING THE LIBRARIES

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.io as pio
```

2.IMPORTING THE DATASET

In [6]:

```
dataset=pd.read_csv('Zomato_Mumbai_Dataset.csv',delimiter='|')
```

In [7]:

```
dataset.head()
```

Out[7]:

	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_T
0	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	Mumbai	First International Financial Centre--Bandra ...	https://www.zomato.com/mumbai/hitchki-bandra-k...	1	Casual Dining	12noon to 130am(Mon-Sun)	Exce
1	Baba Falooda	400	Desserts,Ice Cream,Beverages	Mumbai	Mahim	https://www.zomato.com/mumbai/baba-falooda-mah...	1	Dessert Parlor	2pm to 1am(Mon-Sun)	Very G
2	Chin Chin Chu	1800	Asian,Chinese	Mumbai	Juhu	https://www.zomato.com/mumbai/chin-chin-chu-ju...	1	Casual Dining	12noon to 1am(Mon-Sun)	Very G
3	Butterfly High	1000	Modern Indian	Mumbai	Bandra Kurla Complex	https://www.zomato.com/mumbai/butterfly-high-b...	1	Bar	12noon to 130am(Mon-Sun)	Very G
4	BKC DIVE	1200	North Indian,Chinese,Continental	Mumbai	Bandra Kurla Complex	https://www.zomato.com/mumbai/bkc-dive-bandra-...	1	Bar	1130am to 1am(Mon-Sun)	Veľmi d

3.GETTING BASIC INFORMATION ABOUT THE DATASET

In [8]:

```
dataset.shape
```

Out[8]:

```
(15081, 12)
```

In [9]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15081 entries, 0 to 15080
Data columns (total 12 columns):
NAME                 15081 non-null object
PRICE                15080 non-null object
CUSINE_CATEGORY      15079 non-null object
CITY                 15080 non-null object
REGION               15080 non-null object
URL                  15080 non-null object
PAGE NO              15080 non-null object
CUSINE TYPE          15080 non-null object
TIMING               15015 non-null object
RATING_TYPE          15080 non-null object
RATING               15080 non-null object
VOTES                15080 non-null object
dtypes: object(12)
memory usage: 1.4+ MB
```

In [10]:

```
dataset.describe()
```

Out[10]:

	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
count	15081	15080	15079	15080	15080	15080	15080	15080	15015	15080	15080	15080
unique	12720	67	3183	2	241	13823	944	23	2551	32	35	1124
top	NAME	400	CUSINE_CATEGORY	Mumbai	REGION	URL	PAGE NO	Quick Bites	11am to 11pm(Mon-Sun)	Average	-	-
freq	942	2042	942	14138	942	942	942	5262	1192	5112	2360	2360

4.CLEANING THE DATAET

4.a) Removing the redundunt rows of data

In [11]:

```
# checking redundunt rows of data

wrong_data=dataset['PAGE NO']== 'PAGE NO'
dataset[wrong_data]
```

Out[11]:

	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
15	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
31	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
47	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
63	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
79	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
...	...	...	...	...	...	...	...	...	...	...	...	...
15000	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
15016	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
15032	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
15048	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
15064	NAME	PRICE	CUSINE_CATEGORY	CITY	REGION	URL	PAGE NO	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES

942 rows × 12 columns

In [12]:

```
# performing negation of the wrong dataset and then storing the correct data back in the dataset Dataframe
# this permanently removes the wrong data from the original dataframe

dataset=dataset[~wrong_data]
```

In [13]:

```
# Dropping coloumns which are not required for future analysis

dataset.drop(['URL','PAGE NO','CITY'], axis=1, inplace=True)
```

```
C:\Users\Lenovo\Anaconda3\lib\site-packages\pandas\core\frame.py:4102: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y)
  errors=errors,
```

In [14]:

```
dataset.head()
```

Out[14]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
0	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	First International Financial Centre-- Bandra ...	Casual Dining	12noon to 130am(Mon-Sun)	Excellent	4.9	3529
1	Baba Falooda	400	Desserts,Ice Cream,Beverages	Mahim	Dessert Parlor	2pm to 1am(Mon-Sun)	Very Good	4.4	1723
2	Chin Chin Chu	1800	Asian,Chinese	Juhu	Casual Dining	12noon to 1am(Mon-Sun)	Very Good	4.2	337
3	Butterfly High	1000	Modern Indian	Bandra Kurla Complex	Bar	12noon to 130am(Mon-Sun)	Very Good	4.3	1200
4	BKC DIVE	1200	North Indian,Chinese,Continental	Bandra Kurla Complex	Bar	1130am to 1am(Mon-Sun)	Veľmi dobré	4.4	5995

4.b) Removing the null records

In [16]:

```
# checking for null records

dataset.isnull().sum()
```

Out[16]:

```
NAME                 0
PRICE                1
CUSINE_CATEGORY      2
REGION               1
CUSINE TYPE          1
TIMING              66
RATING_TYPE          1
RATING               1
VOTES                1
dtype: int64
```

In [17]:

```
# checking for a null row

dataset[dataset['PRICE'].isnull()]
```

Out[17]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES
15080		NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [18]:

```
# dropping the above row from the dataset

dataset=dataset.drop(labels=15080, axis=0)
```

In [19]:

```
# Replacing the other null records with NA

dataset.fillna('NA', inplace=True)
```

In [20]:

```
# Confirming all the null records are correct

dataset.isnull().sum()
```

Out[20]:

```
NAME                0
PRICE               0
CUSINE_CATEGORY     0
REGION              0
CUSINE TYPE         0
TIMING              0
RATING_TYPE         0
RATING              0
VOTES               0
dtype: int64
```

4.c) Converting the data types of numerical columns to numerical datatypes

In [21]:

```
# Checking for text values in the column before converting it to numeric datatype

dataset['RATING'].value_counts()
```

Out[21]:

```
-          2360
3.5        1094
3.4        1036
3.6         960
NEW         953
3.3         926
3.7         917
3.2         801
3.8         782
3.1         734
3.0         622
3.9         596
2.9         409
4.0         408
2.8         309
4.1         298
4.2         199
2.7         170
4.3         148
4.4          99
2.6          77
Opening      57
4.5          46
2.5          39
4.6          32
2.4          26
4.7          13
2.3          10
2.1           5
2.2           4
4.8           4
4.9           2
2.0           1
1.8           1
Name: RATING, dtype: int64
```

In [23]:

```
# Replacing the text values with '0'

dataset['RATING'].replace(to_replace=['-','NEW','Opening'] , value='0' , inplace= True)
```

In [24]:

```
# Checking for text values in the column before converting it to numeric datatype

dataset['VOTES'].value_counts()
```

Out[24]:

```
-       2360
NEW      953
4        364
5        320
6        288
        ... 
817        1
3376       1
3283       1
857        1
603        1
Name: VOTES, Length: 1123, dtype: int64
```

In [25]:

```
# Replacing the text values with '0'

dataset['VOTES'].replace(to_replace=['-','NEW','Opening'] , value='0' , inplace= True)
```

In [26]:

```
# Changing the datatypes of the numerical columns

dataset['PRICE']= dataset['PRICE'].astype('int64')
dataset['VOTES']= dataset['VOTES'].astype('int64')
dataset['RATING']= dataset['RATING'].astype('float64')
```

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14138 entries, 0 to 15079
Data columns (total 9 columns):
NAME                  14138 non-null object
PRICE                 14138 non-null int64
CUSINE_CATEGORY       14138 non-null object
REGION                14138 non-null object
CUSINE TYPE           14138 non-null object
TIMING                14138 non-null object
RATING_TYPE           14138 non-null object
RATING                14138 non-null float64
VOTES                 14138 non-null int64
dtypes: float64(1), int64(2), object(6)
memory usage: 1.1+ MB
```

4.d) Working with Timing column

In [28]:
```
dataset['TIMING'].value_counts()
```

Out[28]:
```
11am to 11pm(Mon-Sun)                                          1192
11am to 12midnight(Mon-Sun)                                     632
12noon to 12midnight(Mon-Sun)                                   467
11am to 1130pm(Mon-Sun)                                         309
10am to 10pm(Mon-Sun)                                           267
                                                               ... 
4pm to 230am(Mon-Sun)                                             1
10am to 10pm(Mon-Fri),Closed(Sat-Sun)                             1
7am to 11pm(Mon,Tue,Thu,Fri,Sat,Sun),Closed...                    1
Closed(Mon,Thu,Fri),1230pm to 1130pm(Tue...                       1
8am to 9pm(Mon,Tue,Wed,Fri,Sat,Sun),Closed...                     1
Name: TIMING, Length: 2551, dtype: int64
```

In [29]:

```
# Splitting the column and storing it in temp_df dataframe

temp_df = dataset['TIMING'].str.split("(" , n=1 , expand= True )
temp_df
```

Out[29]:

	0	1
0	12noon to 130am	Mon-Sun)
1	2pm to 1am	Mon-Sun)
2	12noon to 1am	Mon-Sun)
3	12noon to 130am	Mon-Sun)
4	1130am to 1am	Mon-Sun)
...	...	...
15075	8am to 11pm,12midnight to 115am	Mon-Sun)
15076	11am to 230am	Mon-Sun)
15077	11am to 11pm	Mon,Tue,Wed,Thu,Sun),11am to ...
15078	9am to 1230AM	Mon-Sun)
15079	12noon to 330pm,7pm to 1am	Mon-Sun)

14138 rows × 2 columns

In [30]:

```
# Assigning the columns back to the dataset Dataframe

dataset['TIMING']= temp_df[0]
dataset['DAYS OPEN']= temp_df[1]
dataset.head()
```

Out[30]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
0	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	First International Financial Centre-- Bandra ...	Casual Dining	12noon to 130am	Excellent	4.9	3529	Mon-Sun)
1	Baba Falooda	400	Desserts,Ice Cream,Beverages	Mahim	Dessert Parlor	2pm to 1am	Very Good	4.4	1723	Mon-Sun)
2	Chin Chin Chu	1800	Asian,Chinese	Juhu	Casual Dining	12noon to 1am	Very Good	4.2	337	Mon-Sun)
3	Butterfly High	1000	Modern Indian	Bandra Kurla Complex	Bar	12noon to 130am	Very Good	4.3	1200	Mon-Sun)
4	BKC DIVE	1200	North Indian,Chinese,Continental	Bandra Kurla Complex	Bar	1130am to 1am	Veľmi dobré	4.4	5995	Mon-Sun)

In [32]:

```
# Removing the bracket character from the days column

dataset['DAYS OPEN']= dataset['DAYS OPEN'].str.replace(')', "" , regex= True)
dataset.head()
```

Out[32]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
0	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	First International Financial Centre-- Bandra ...	Casual Dining	12noon to 130am	Excellent	4.9	3529	Mon-Sun
1	Baba Falooda	400	Desserts,Ice Cream,Beverages	Mahim	Dessert Parlor	2pm to 1am	Very Good	4.4	1723	Mon-Sun
2	Chin Chin Chu	1800	Asian,Chinese	Juhu	Casual Dining	12noon to 1am	Very Good	4.2	337	Mon-Sun
3	Butterfly High	1000	Modern Indian	Bandra Kurla Complex	Bar	12noon to 130am	Very Good	4.3	1200	Mon-Sun
4	BKC DIVE	1200	North Indian,Chinese,Continental	Bandra Kurla Complex	Bar	1130am to 1am	Veľmi dobré	4.4	5995	Mon-Sun

In [34]:

```
# checking for null records in DAYS OPEN column

dataset['DAYS OPEN'].isnull().sum()
```

Out[34]:

```
160
```

In [35]:

```
# Replacing the null values with NA

dataset.fillna("NA" , inplace= True)
```

In [36]:

```
# Checking info of all the columns

dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14138 entries, 0 to 15079
Data columns (total 10 columns):
NAME                  14138 non-null object
PRICE                 14138 non-null int64
CUSINE_CATEGORY       14138 non-null object
REGION                14138 non-null object
CUSINE TYPE           14138 non-null object
TIMING                14138 non-null object
RATING_TYPE           14138 non-null object
RATING                14138 non-null float64
VOTES                 14138 non-null int64
DAYS OPEN             14138 non-null object
dtypes: float64(1), int64(2), object(7)
memory usage: 1.2+ MB
```

4.e) Removing the restaurant records whose rating or Votes is 0

In [40]:

```python
# Finding those restaurants whose rating and votes are Zero

useless_data= (dataset['RATING']==0.0) | (dataset['VOTES']== 0)
dataset[useless_data]
```

Out[40]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
32	Hotel Annapoorna Refreshments	400	Maharashtrian,Mughlai,Chinese	Ghansoli	Quick Bites	1030am to 1230AM	Not rated	0.0	0	Mon-Sun
34	Biryani 9	600	Biryani,North Indian	Near Andheri East Station	none	11am to 3am	None	0.0	0	Mon-Sun
36	D Fusion Flavours	350	Chinese	Goregaon East	none	12noon to 330pm,7pm to 3am	None	0.0	0	Mon-Sun
39	Nation Tadka	400	North Indian,South Indian,Chinese,Fast Food	Worli	none	12noon to 1230AM	Not rated	0.0	0	Mon-Sun
83	Link Way Restaurant	500	North Indian,Chinese	Jogeshwari	Quick Bites	12noon to 4pm,8pm to 1am	Not rated	0.0	0	Mon-Sun
...	...	...	...	...	...	...	...	...	...	...
14998	Foodies House	0	Chinese	Goregaon East	none	12noon to 4am	None	0.0	0	Mon-Sun
14999	Khansama	0	Biryani	Lower Parel	none	12noon to 3am	None	0.0	0	Mon-Sun
15010	Earth Cafe @ Waterfield	800	Cafe,Healthy Food,Italian,Pizza,Beverages	Linking Road-- Bandra West	Café	10am to 10pm	None	0.0	0	Mon-Thu,10am to 11pm(Fri-Sun
15023	How About Some Cream	200	Beverages	Mumbai Central	Beverage Shop	12noon to 3am	None	0.0	0	Mon-Sun
15046	Food And Taste Theory	800	Continental,Italian	Phoenix Marketcity-- Kurla	Casual Dining	9am to 12midnight	None	0.0	0	Mon-Fri,9am to 1230AM(Sat-Sun

3371 rows × 10 columns

In [41]:

```
## Performing negation of the useless dataset and then storing the correct data back in the dataset Dataframe
## this permanently removes the wrong data from the original dataframe

dataset= dataset[~useless_data]
```

4.f) Working on "RATING TYPE" Column

In [42]:

```
# checking the unique values in the column

dataset['RATING_TYPE'].value_counts()
```

Out[42]:

```
Average          5111
Good             4330
Very Good        1137
Excellent          95
Poor               47
Veľmi dobré         6
Skvělá volba        4
Dobrze              4
Bardzo dobrze       3
Dobré               2
Bueno               2
Buono               2
Priemer             2
Bom                 2
İyi                 2
Ortalama            2
Promedio            2
Průměr              2
Muito Bom           2
Skvělé              1
Baik                1
Sangat Baik         1
Muy Bueno           1
Media               1
Çok iyi             1
Velmi dobré         1
Biasa               1
Média               1
Excelente           1
Name: RATING_TYPE, dtype: int64
```

In [73]:

```
# Translating texts into proper english text

dataset['RATING_TYPE'].replace(to_replace='Excelente' , value ='Excellent', inplace= True)

dataset['RATING_TYPE'].replace(to_replace=['Veľmi dobré','Bardzo dobrze', 'Muy Bueno', 'Velmi dobré'], value ='Very Good

dataset['RATING_TYPE'].replace(to_replace=['Skvělá volba','Dobrze', 'Bueno', 'Buono','Dobré','Bom','Skvělé'] ,value ='Go

dataset['RATING_TYPE'].replace(to_replace=['Priemer','Média', 'Çok iyi'] ,value ='Average', inplace= True)

dataset['RATING_TYPE'].replace(to_replace=['Průměr','Promedio', 'Ortalama', 'Muito Bom','İyi'] ,value ='Poor', inplace=

dataset['RATING_TYPE'].replace(to_replace=['Baik','Biasa', 'Media', 'Sangat Baik'] ,value ='Very Poor', inplace= True)
```

In [74]:

```
# Checking all the values correctly mapped

dataset['RATING_TYPE'].value_counts()
```

Out[74]:

```
Average      5115
Good         4347
Very Good    1148
Excellent      96
Poor           57
Very Poor       4
Name: RATING_TYPE, dtype: int64
```

4.g) Working on REGION Column

In [75]:
```
dataset['REGION'].value_counts()
```

Out[75]:
```
Mira Road                        405
Malad West                       308
Chembur                          277
Kharghar                         268
Borivali West                    264
                                ... 
Le Sutra Hotel-- Khar              1
Yogi Midtown-- Turbhe              1
Hotel Mirador-- Andheri East       1
Hotel Jawahar-- Ulhasnagar         1
Cooling Tower-- Goregaon East      1
Name: REGION, Length: 237, dtype: int64
```

In [76]:
```
# Removing the irrelavant text from the region column

dataset['REGION']= dataset['REGION'].str.replace('[a-zA-Z].+--','', regex= True)
```

In [79]:
```
# Removing the west and east from the region column

dataset['REGION']= dataset['REGION'].str.replace(' West| west| East| east','', regex= True)
```

In [80]:
```
dataset['REGION'].value_counts()
```

Out[80]:
```
 Thane                 712
 Andheri               407
Mira Road              405
Malad                  308
 Bandra                282
                      ... 
 Chakala                 1
 Ulhasnagar              1
Girgaon Chowpatty        1
 Worli                   1
 Bhayandar               1
Name: REGION, Length: 163, dtype: int64
```

In [81]:
```
# Replacing the small regions with known region names

dataset['REGION']= dataset['REGION'].str.replace('4 Bungalows| 7 Anderi| Azad Nagar| Near Andheri station| Veera Desai A
dataset['REGION'] = dataset['REGION'].str.replace('Bandra Kurla Complex','Bandra',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('CBD-Belapur','CBD Belapur',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('Girgaon Chowpatty','Chowpatty',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('Dadar Shivaji Park','Dadar',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('Flea Bazaar Café|Kamala Mills Compound','Lower Parel',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('Runwal Green','Mulund',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('Mumbai CST Area','Mumbai Central',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('Kopar Khairane|Seawoods|Turbhe|Ulwe','Navi Mumbai',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('New Panvel|Old Panvel','Panvel',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('Kamothe','Sion',regex=True)
dataset['REGION'] = dataset['REGION'].str.replace('Ghodbunder Road|Majiwada','Thane',regex=True)
```

4.h) Removing Duplicate Records

In [82]:

```
# Finding all the duplicate rows

dataset[dataset.duplicated()]
```

Out[82]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
4064	Sai Sannidhi Restaurant & Bar	1000	North Indian,Konkan	Dahisar	Casual Dining	11am to 12midnight	Good	3.7	99	Mon-Sun
4068	Konkan Katta	400	Seafood,Maharashtrian,Malwani	Mahakali	Quick Bites	11am to 330pm,630pm to 1130pm	Good	3.5	181	Mon-Sun
4082	Usmaniya Hotel	600	Mughlai	Fort	Casual Dining	1030am to 1130pm	Average	3.2	8	Mon-Sun
4083	Gina's Cakes	450	Bakery	Dombivali	none	11am to 11pm	Good	3.5	49	Mon-Sun
4084	Konkanastha Lunch Home	400	Seafood,Malwani	Chakala	Casual Dining	12noon to 3pm,730pm to 1030pm	Good	3.5	44	Mon-Sun
...	...	...	...	...	...	...	...	...	...	...
14200	Mezbaan Family Restaurant	350	Chinese,Mughlai	Mumbra	Dhaba	12noon to 1230AM	Average	2.8	97	Mon-Sun
14204	Jyoti Lunch Home	650	Chinese,North Indian,Seafood,Mughlai	Mulund	Casual Dining	11am to 1230AM	Good	3.5	49	Mon-Sun
14253	On Toes	900	Italian,North Indian,Chinese	Malad	Casual Dining	12noon to 3pm,7pm to 1230AM	Good	3.6	76	Mon-Sun
14761	Frosty Farm	400	Ice Cream,Desserts,Fast Food	Malad	Dessert Parlor	1pm to 1215AM	Good	3.6	120	Mon-Sun
14928	Shree Manu Sagar	300	North Indian,Chinese,Indian	Ghansoli	Quick Bites	1130am to 415pm,7pm to 1215AM	Average	3.3	51	Mon-Sun

220 rows × 10 columns

In [83]:
```
# Dropping all the duplicate rows

dataset= dataset.drop_duplicates()
```

5. COPYING THE CLEANED DATA INTO A NEW DATAFRAME

In [84]:
```
zomato_df= dataset.copy()
```

In [85]:
```
zomato_df.head()
```

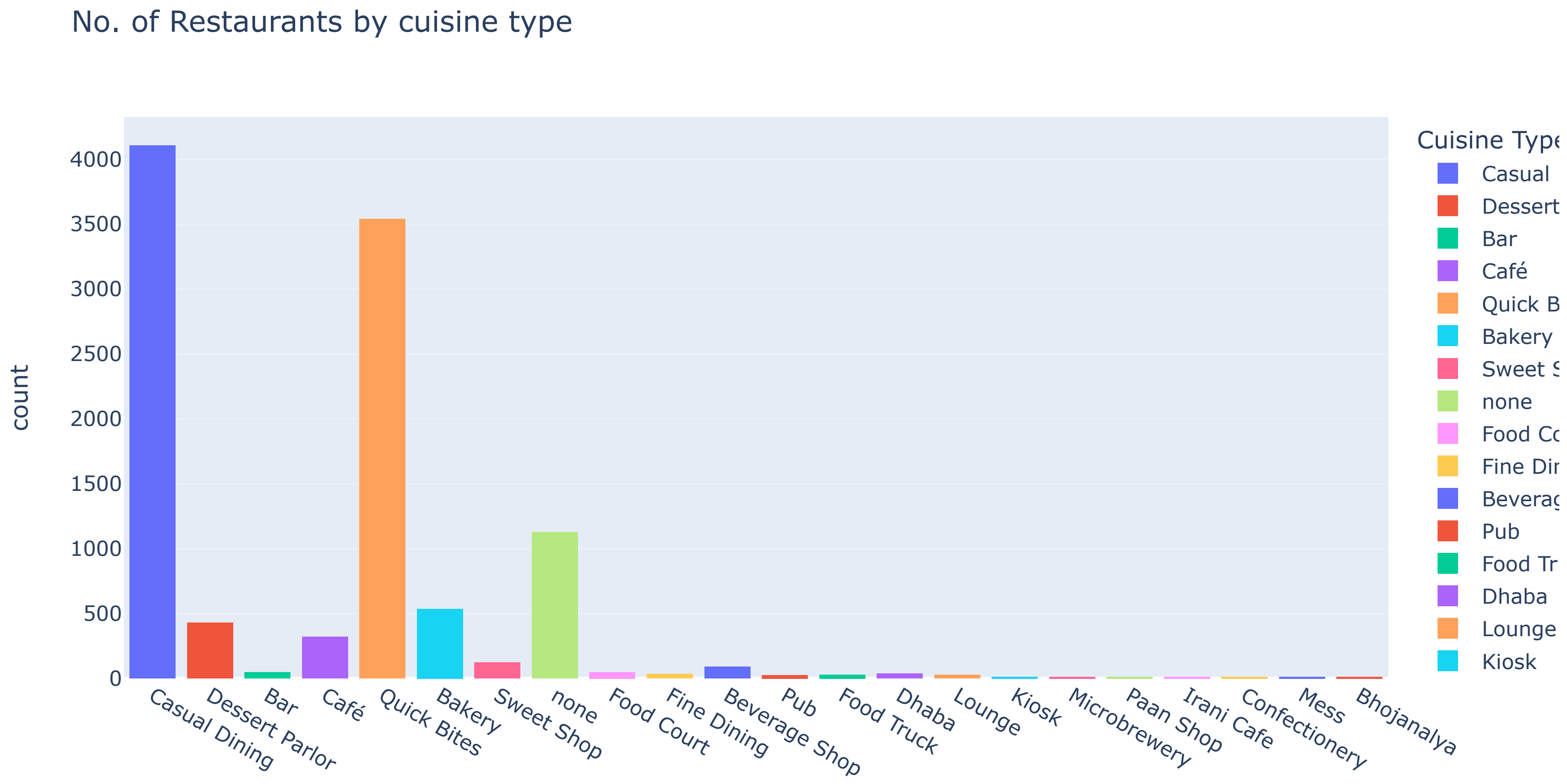
Out[85]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
0	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	Bandra	Casual Dining	12noon to 130am	Excellent	4.9	3529	Mon-Sun
1	Baba Falooda	400	Desserts,Ice Cream,Beverages	Mahim	Dessert Parlor	2pm to 1am	Very Good	4.4	1723	Mon-Sun
2	Chin Chin Chu	1800	Asian,Chinese	Juhu	Casual Dining	12noon to 1am	Very Good	4.2	337	Mon-Sun
3	Butterfly High	1000	Modern Indian	Bandra	Bar	12noon to 130am	Very Good	4.3	1200	Mon-Sun
4	BKC DIVE	1200	North Indian,Chinese,Continental	Bandra	Bar	1130am to 1am	Very Good	4.4	5995	Mon-Sun

6. PERFORMING EXPLORATORY DATA ANALYSIS

Q1) How many restaurants are there in mumbai for each type of cusine?

```
In [86]: fig = px.histogram(zomato_df , x='CUSINE TYPE', color='CUSINE TYPE', title='No. of Restaurants by cuisine type', labels=
         fig.show()
```



Q2) What are the percentage of restaurants by rating type in mumbai?

In [87]:

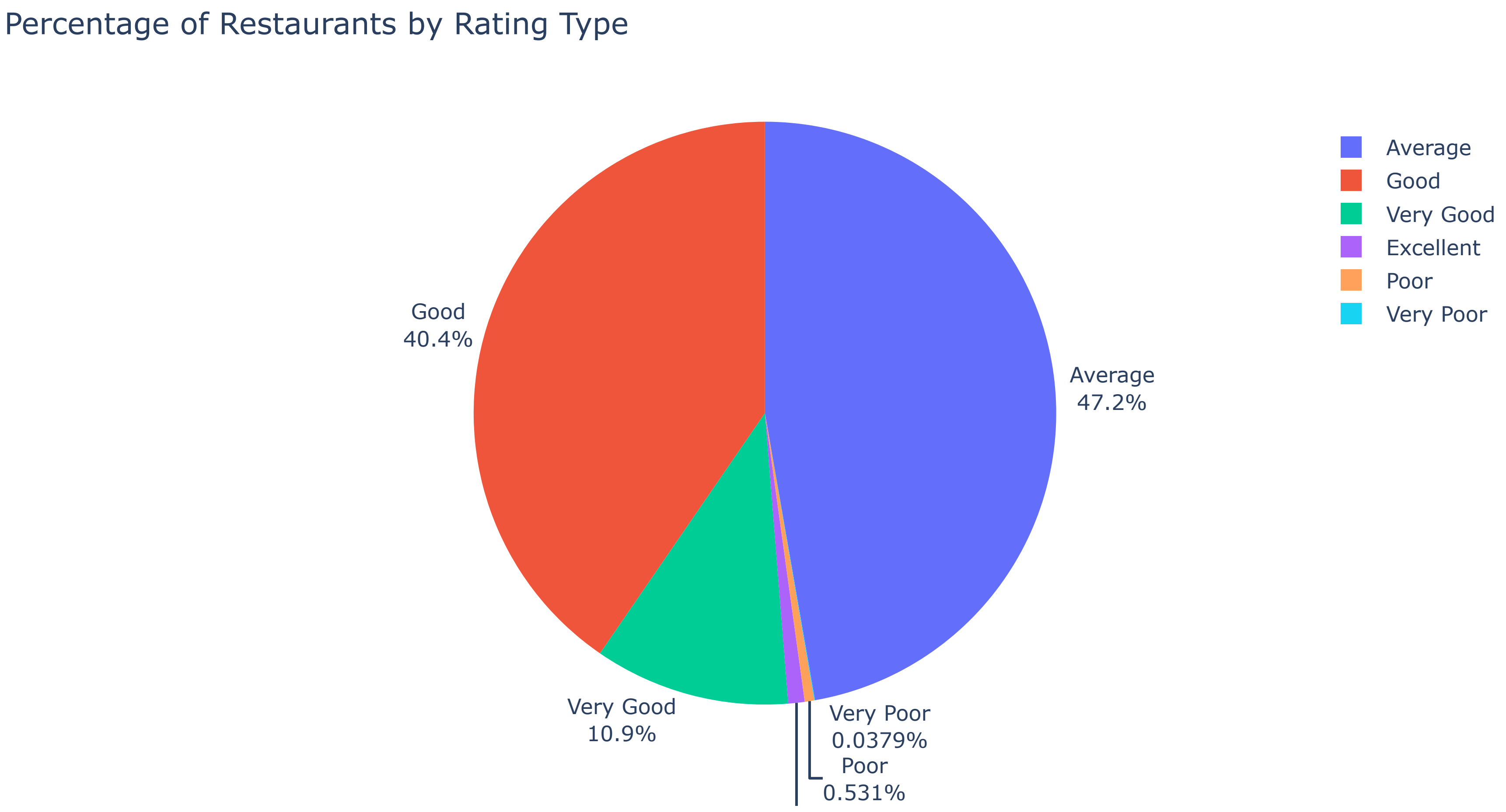
```
rating_type_df= zomato_df['RATING_TYPE'].value_counts().reset_index()
rating_type_df.rename(columns={'index':'RATING TYPE', 'RATING_TYPE':'COUNT OF RESTAURANTS'}, inplace=True)
rating_type_df
```

Out[87]:

	RATING TYPE	COUNT OF RESTAURANTS
0	Average	4983
1	Good	4263
2	Very Good	1145
3	Excellent	96
4	Poor	56
5	Very Poor	4

```
In [89]: fig = px.pie(rating_type_df, names='RATING TYPE', values='COUNT OF RESTAURANTS', color='RATING TYPE',
                title='Percentage of Restaurants by Rating Type').update_traces(textposition='outside', textinfo='percent+label')

         fig.show()
```



Q3)Which are the Top 10 highest rated Seafood Restaurant in Mumbai?

In [90]:

```
seafood_df = zomato_df[zomato_df['CUSINE_CATEGORY'].str.contains('Seafood')]
seafood_df.sort_values(by='RATING',ascending=False).head(10)
```

Out[90]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
7104	Thangabali	1000	Seafood,South Indian,Mangalorean,Andhra,Kerala	Khar	Bar	12noon to 4pm,7pm to 3am	Excellent	4.7	564	Mon-Sun
76	Ceremonial Kitchen & Co	1000	Seafood,Maharashtrian,North Indian,Chinese	Thane	Casual Dining	1130am to 1130pm	Excellent	4.6	350	Mon-Sun
13685	Maharashtra Lunch Home	600	Maharashtrian,Malwani,Konkan,Seafood	Kharghar	Casual Dining	11am to 345pm,7pm to 1145pm	Excellent	4.6	209	Mon-Sun
12433	Quarter Canteen	1100	North Indian,Seafood,Chinese	Bandra	Casual Dining	12noon to 330pm,7pm to 1am	Excellent	4.5	573	Mon-Sun
902	The Harbour Bay - SeaFood Kitchen & Bar	2400	Seafood,Beverages	Bandra	Casual Dining	12noon to 1am	Excellent	4.5	100	Mon-Sun
884	Rajmanya-Seafood family restaurant	800	Maharashtrian,Konkan,Seafood	Vashi	Casual Dining	11am to 11pm	Excellent	4.5	178	Mon-Sun
3380	Peco Peco	700	Chinese,Seafood,Asian	Powai	none	12noon to 330pm,7pm to 1230AM	Excellent	4.5	497	Mon-Sun
9954	Pi Bar and Kitchen	1600	Continental,European,Italian,Seafood,Pizza,Des...	Andheri	Bar	12noon to 6pm,7pm to 12midnight	Excellent	4.5	2068	Mon-Sun
903	Ferry Wharf	1500	Seafood,Mangalorean	Bandra	Casual Dining	11am to 330pm,7pm to 1230AM	Very Good	4.4	459	Mon-Sun
915	Monis Bar and Restaurant	1000	North Indian,Chinese,Continental,Seafood,Bever...	Thane	Casual Dining	1130am to 330pm,6pm to 1130pm	Very Good	4.4	662	Mon-Sun

Q4) Which is the best Food Truck in Mumbai?

In [91]:

```
foodtruck_df = zomato_df[zomato_df['CUSINE TYPE'] == 'Food Truck']
foodtruck_df.sort_values(by='RATING',ascending=False).head(2)
```

Out[91]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
262	Dumpling Delights	200	Momos	Matunga	Food Truck	430pm to 930pm	Very Good	4.3	212	Mon-Sun
1017	Street Food Co.	250	Fast Food,Chinese	Virar	Food Truck	6pm to 3am	Very Good	4.1	274	Mon-Sun

Q5)Which places have the highest rated restaurant for each Cuisine Type in Mumbai?

In [92]:

```
# Assuming restaurants having rating above 4.5

highest_rated_df = zomato_df[zomato_df['RATING'] >= 4.5]
highest_rated_df
```

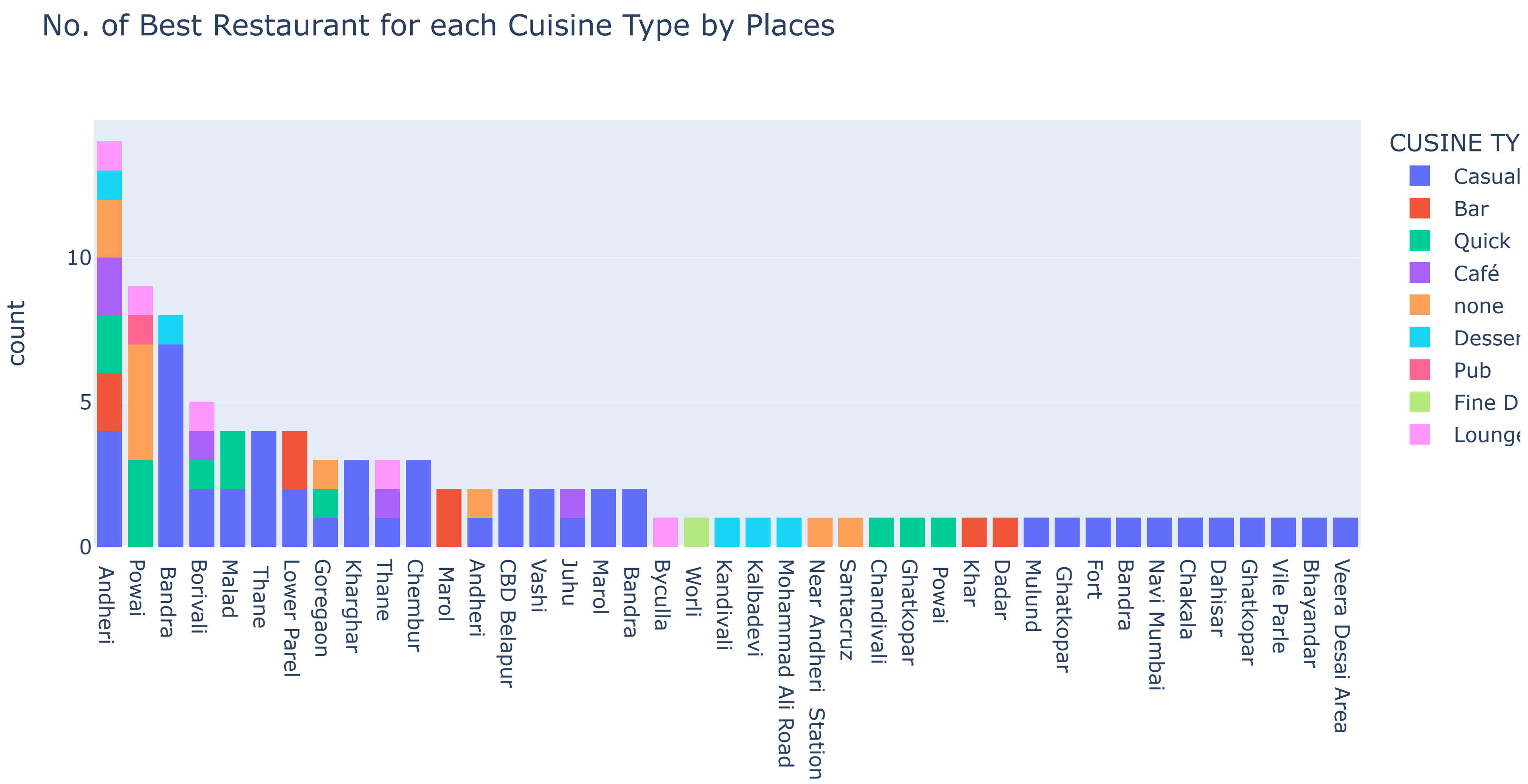
Out[92]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
0	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	Bandra	Casual Dining	12noon to 130am	Excellent	4.9	3529	Mon-Sun
6	Persian Darbar	1300	Biryani,North Indian,Chinese,Mughlai	Marol	Casual Dining	10am to 3am	Excellent	4.5	3058	Mon-Sun
7	Tanatan	1500	Modern Indian	Juhu	Casual Dining	12noon to 130am	Excellent	4.7	1842	Mon-Sun
9	Plum by Bent Chair	1800	Asian	Lower Parel	Casual Dining	12noon to 1am	Excellent	4.7	1876	Mon-Sun
10	Angrezi Dhaba	1500	North Indian,Chinese,Thai,European	Dadar	Bar	12noon to 1am	Excellent	4.5	2092	Mon-Sun
...	...	...	...	...	...	...	...	...	...	...
14228	Zaika Crave - Club Aquaria	1300	North Indian,Continental,Chinese,Desserts	Borivali	Casual Dining	11am to 330pm,7pm to 1130pm	Excellent	4.5	1302	Mon,Tue,Wed...
14234	Cone Culture	250	European	Kharghar	Casual Dining	Closed	Excellent	4.6	492	Mon,12noon to 11pm(Tue-Sun
15007	Dessertino	300	Desserts,Ice Cream	Kandivali	Dessert Parlor	11am to 12midnight	Excellent	4.8	184	Mon-Sun
15051	Tick-eat	800	North Indian,Italian,Chinese,Mexican,Lebanese	Mulund	Casual Dining	1130am to 330pm,7pm to 1130pm	Excellent	4.5	754	Mon-Sun
15056	Daftar Goregaon	750	Pizza,Chinese,North Indian,Beverages	Goregaon	Casual Dining	12noon to 1130pm	Excellent	4.6	388	Mon-Sun

97 rows × 10 columns

In [94]:
```
fig = px.histogram(highest_rated_df, x='REGION', color='CUSINE TYPE',
          title= 'No. of Best Restaurant for each Cuisine Type by Places').update_xaxes(categoryorder="total descendi

fig.show()
```



Q6) What is the Avg Price Distibution of highest rated restaurant for each Cuisine Type in Mumbai?

In [95]:

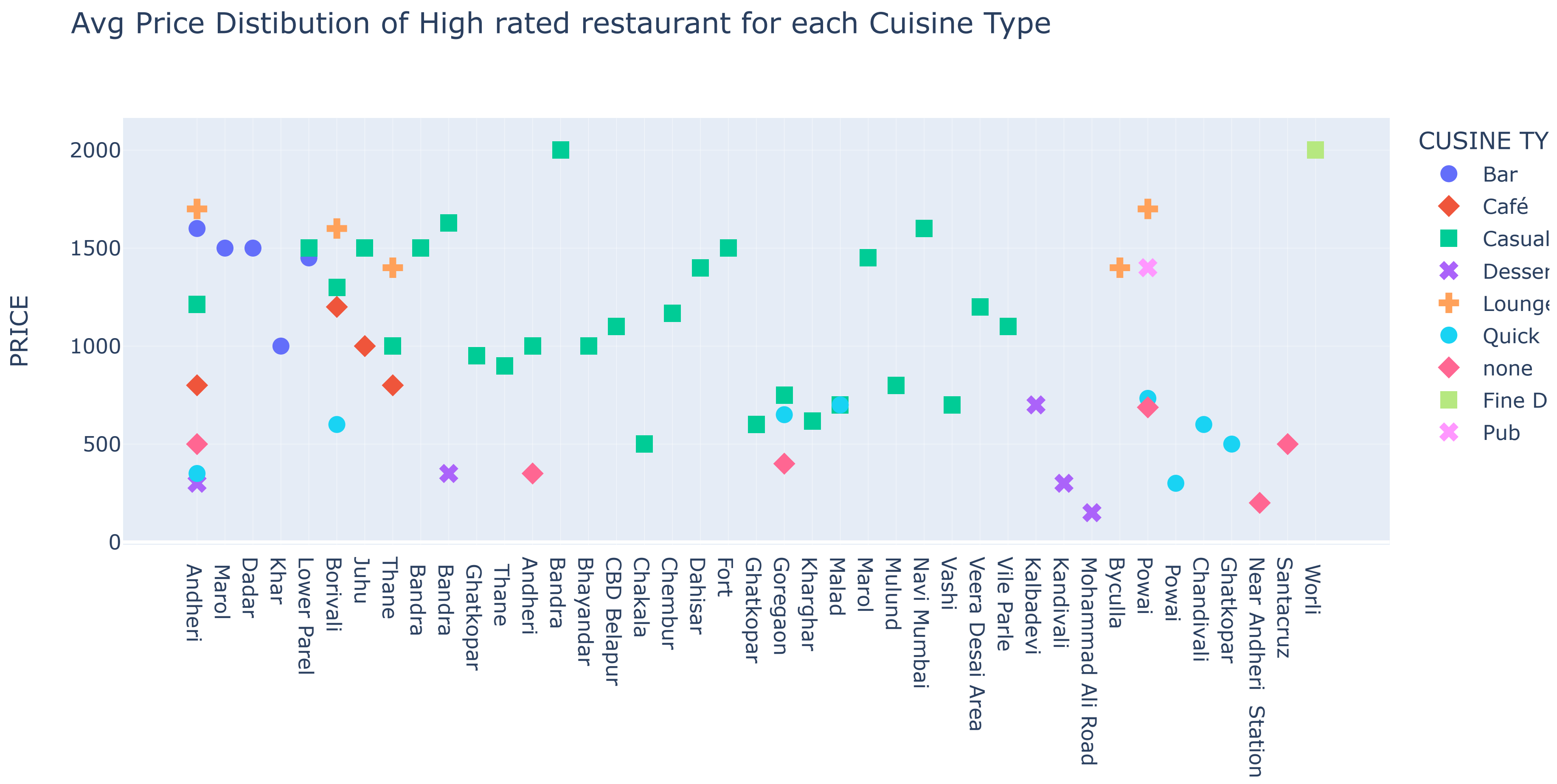
```
highest_rated_price_df = highest_rated_df.groupby(by=['REGION', 'CUSINE TYPE'])['PRICE'].mean().reset_index()
highest_rated_price_df.head()
```

Out[95]:

	REGION	CUSINE TYPE	PRICE
0	Andheri	Bar	1600.0
1	Andheri	Café	800.0
2	Andheri	Casual Dining	1212.5
3	Andheri	Dessert Parlor	300.0
4	Andheri	Lounge	1700.0

In [101]:
```
fig = px.scatter(highest_rated_price_df, x="REGION", y="PRICE", color="CUSINE TYPE", symbol="CUSINE TYPE",
                 title=' Avg Price Distibution of High rated restaurant for each Cuisine Type').update_traces(marker_size=10)

fig.show()
```



Q7) Which areas have a large number of Chinese Restaurant Market?

In [102]:
```
chinese_df = zomato_df[zomato_df['CUSINE_CATEGORY'].str.contains('Chinese')]
chinese_df
```

Out[102]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	TIMING	RATING_TYPE	RATING	VOTES	DAYS OPEN
0	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	Bandra	Casual Dining	12noon to 130am	Excellent	4.9	3529	Mon-Sun
2	Chin Chin Chu	1800	Asian,Chinese	Juhu	Casual Dining	12noon to 1am	Very Good	4.2	337	Mon-Sun
4	BKC DIVE	1200	North Indian,Chinese,Continental	Bandra	Bar	1130am to 1am	Very Good	4.4	5995	Mon-Sun
5	Flea Bazaar Café	800	American,Asian,Street Food,North Indian,Luckno...	Lower Parel	Café	12noon to 1am	Very Good	4.2	2042	Mon-Sun
6	Persian Darbar	1300	Biryani,North Indian,Chinese,Mughlai	Marol	Casual Dining	10am to 3am	Excellent	4.5	3058	Mon-Sun
...	...	...	...	...	...	...	...	...	...	...
15071	Lucknow Zaika	500	North Indian,Chinese	Kurla	Quick Bites	12noon to 2am	Average	2.6	36	Mon-Sun
15072	Zuha's Kitchen	400	Chinese,North Indian,Mughlai	Mumbai Central	Quick Bites	12noon to 4pm,730pm to 430am	Average	3.3	13	Mon-Sun
15075	Tirupati Balaji	500	Chinese,Fast Food,North Indian	Andheri	Casual Dining	8am to 11pm,12midnight to 115am	Good	3.5	267	Mon-Sun
15076	Hari Om Snack Bar	350	Fast Food,South Indian,Chinese	Kandivali	Quick Bites	11am to 230am	Good	3.7	64	Mon-Sun
15079	Mandarin Panda	400	Desserts,Chinese,Thai	Malad	none	12noon to 330pm,7pm to 1am	Good	3.7	121	Mon-Sun

5119 rows × 10 columns

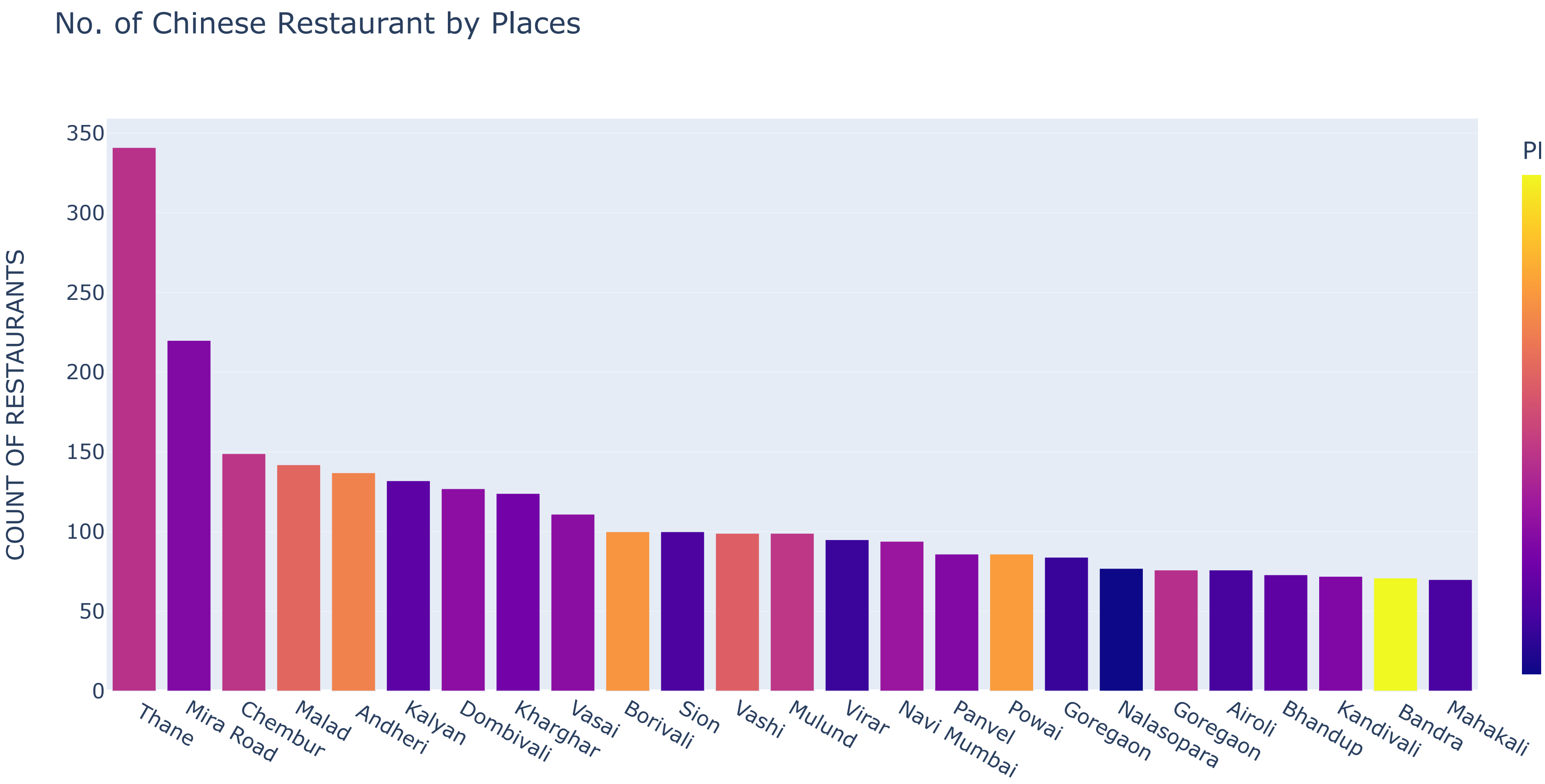
In [107]:
```
chinese_rest_df = chinese_df.groupby(by='REGION').agg({'NAME' : 'count', 'PRICE' : 'mean'}).rename(columns= {'NAME' : 'C
chinese_rest_df = chinese_rest_df.sort_values('COUNT OF RESTAURANTS', ascending=False).head(25)
chinese_rest_df.head()
```

Out[107]:

	REGION	COUNT OF RESTAURANTS	PRICE
35	Thane	341	589.017595
102	Mira Road	220	545.681818
57	Chembur	149	592.617450
95	Malad	142	635.563380
0	Andheri	137	658.394161

In [108]:
```
fig = px.bar(chinese_rest_df, x='REGION', y='COUNT OF RESTAURANTS', color='PRICE', title= 'No. of Chinese Restaurant by

fig.show()
```



Q8) Is there a relation between Price and Rating by each Cuisine Type?

In [109]:

```
price_rating_df = zomato_df.groupby(['CUSINE TYPE', 'RATING'])['PRICE'].mean().reset_index()
price_rating_df
```

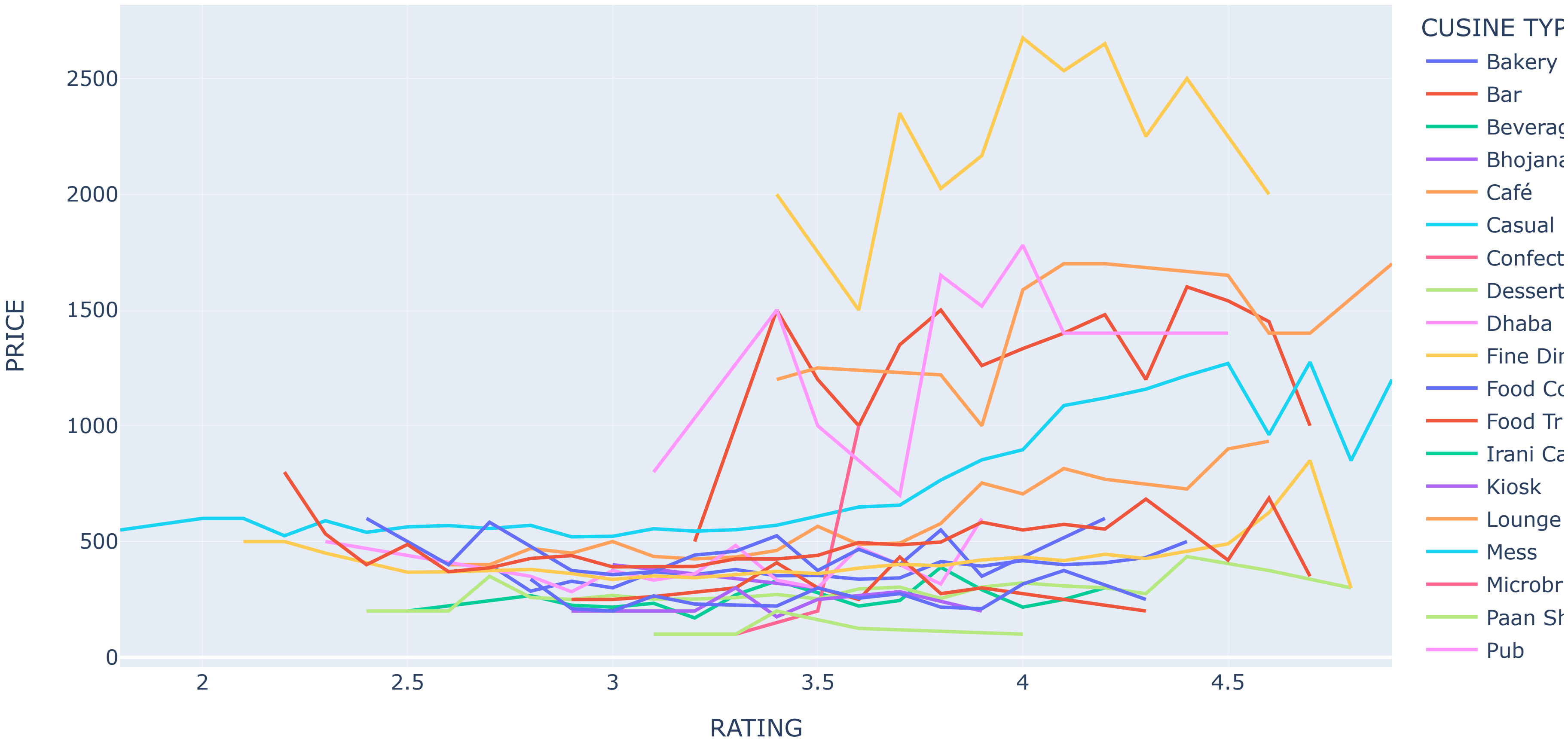
Out[109]:

	CUSINE TYPE	RATING	PRICE
0	Bakery	2.7	400.000000
1	Bakery	2.8	285.714286
2	Bakery	2.9	328.571429
3	Bakery	3.0	300.000000
4	Bakery	3.1	369.117647
...	...	...	...
278	none	4.3	683.333333
279	none	4.4	555.000000
280	none	4.5	420.000000
281	none	4.6	687.500000
282	none	4.7	350.000000

283 rows × 3 columns

In [110]:
```
fig = px.line(price_rating_df, y="PRICE", x="RATING",color='CUSINE TYPE')

fig.show()
```



Q9) Is there a relation between Region and Price?

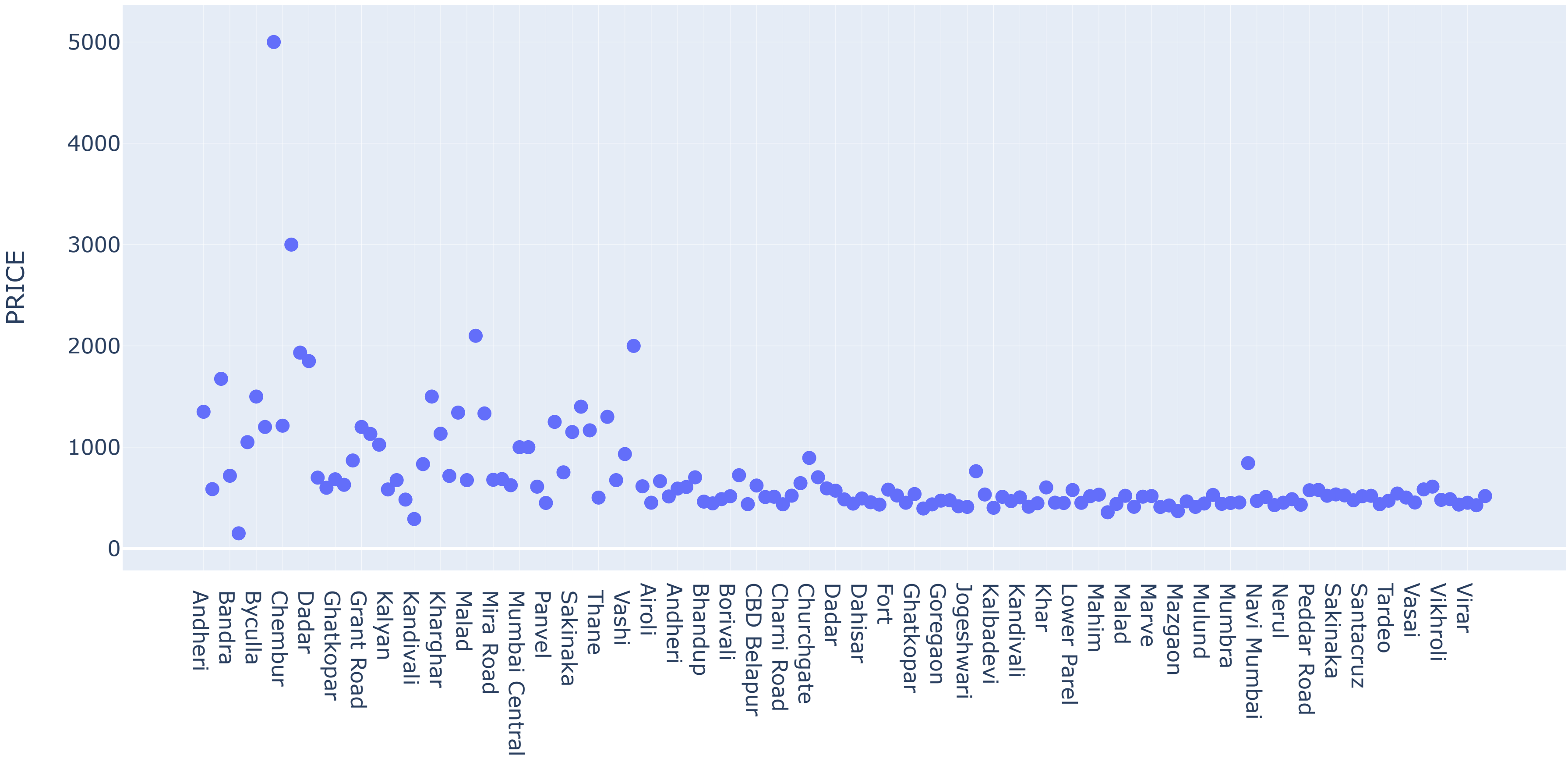
In [111]:
```
region_price_df = zomato_df.groupby(['REGION'])['PRICE'].mean().reset_index()
region_price_df
```

Out[111]:
```
         REGION        PRICE
  0     Andheri  1350.000000
  1     Andheri   586.277916
  2      Bandra  1675.000000
  3      Bandra   718.273381
  4  Bhayandar    150.000000
...         ...          ...
142  Vile Parle   487.368421
143  Vile Parle   433.802817
144       Virar   452.027027
145      Wadala   427.500000
146       Worli   517.763158

147 rows × 2 columns
```

```
In [112]: fig = px.scatter(region_price_df, x="REGION", y="PRICE").update_traces(marker_size=8)

          fig.show()
```



Q10) Find the list of Affordable Restaurants?

Q10) Find the list of Affordable Restaurants

The criteria for Affordable Restaurants would be:- 1) Low Price 2) High Rated

First step will be to find the restaurants with average cost 1/4th the average cost of most expensive restaurant in our dataframe.

Let me explain:-The most expensive restaurant has an average meal cost= 6000. We'll try to stay economical and only pick the restaurants that are 1/4th of 6000.

In [114]:
```
max_price = zomato_df['PRICE'].max()
one_fourth_price = max_price/4
one_fourth_price
```

Out[114]:
```
1250.0
```

In [115]:

```
# Finding list of restaurants that have price less than and equal to 1/4th of the max price i.e Finding Cheap Restaurant

aff_rest_df = zomato_df[['NAME', 'PRICE', 'CUSINE_CATEGORY', 'REGION', 'CUSINE TYPE']]
aff_rest_df = aff_rest_df[aff_rest_df['PRICE'] <= 1250]
aff_rest_df.sort_values(by='PRICE', inplace=True)
aff_rest_df
```

Out[115]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE
6137	Sanjog Wine N Dine	5	North Indian,Chinese	Thane	Casual Dining
2925	Jab We Eat	50	South Indian,North Indian,Maharashtrian,Fast Food	Girgaum	none
9598	Ho5 Store	50	Fast Food	Matunga	none
9589	Golden Butterfly	100	Bakery,Desserts	Mira Road	Bakery
5916	Madhuri Puranpoli	100	Maharashtrian	Vile Parle	none
...	...	...	...	...	...
2740	Peninsula Next	1200	North Indian,Mughlai,Chinese	Sion	Casual Dining
5528	The Thekka	1200	Finger Food,Continental,North Indian,Chinese	Vashi	Lounge
964	Bijoli Grill	1250	Bengali	Powai	Casual Dining
6045	Fabelle at The Chocolate Boutique - ITC Grand ...	1250	Desserts	Parel	Dessert Parlor
7301	SamBar Pub & Kitchen	1250	Finger Food,South Indian,North Indian	Khar	Pub

10190 rows × 5 columns

In [116]:

```
# Finding the highest rated list of restaurants

highrate_rest_df = zomato_df[['NAME', 'PRICE', 'CUSINE_CATEGORY', 'REGION', 'CUSINE TYPE','RATING']]
highrate_rest_df = highrate_rest_df[highrate_rest_df['RATING'] >= 4.5]
highrate_rest_df.sort_values(by='PRICE', inplace=True)
highrate_rest_df
```

Out[116]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	RATING
1502	Cake Centre-The Dessert Maker	150	Desserts	Mohammad Ali Road	Dessert Parlor	4.6
763	Curry And Combos Twist	200	North Indian,Chinese	Andheri	Quick Bites	4.5
807	Moussestruck	200	Desserts	Near Andheri Station	none	4.5
14234	Cone Culture	250	European	Kharghar	Casual Dining	4.6
725	Belo Pops	300	Ice Cream,Desserts,Beverages	Andheri	none	4.5
...	...	...	...	...	...	...
5335	Mia Cucina	2000	Italian	Bandra	Casual Dining	4.5
1786	Global Fusion	2000	Chinese,Japanese,Asian,North Indian	Worli	Fine Dining	4.6
8887	Drifters Tap Station	2000	North Indian,Continental,European,American	Bandra	Casual Dining	4.5
12625	House of Mandarin	2100	Chinese,Sushi,Asian	Bandra	Casual Dining	4.5
902	The Harbour Bay - SeaFood Kitchen & Bar	2400	Seafood,Beverages	Bandra	Casual Dining	4.5

97 rows × 6 columns

Now, we'll merge the aff_rest_df with highrate_rest_df to obtain the intersection i.e the list of Affordable Restaurants !!

In [117]:

```
highrate_aff_df = pd.merge(aff_rest_df, highrate_rest_df, how='inner', on=['NAME', 'REGION'])
highrate_aff_df = highrate_aff_df[['NAME', 'PRICE_x', 'CUSINE_CATEGORY_x', 'REGION', 'CUSINE TYPE_x']]
highrate_aff_df.rename(columns={'NAME':'NAME', 'PRICE_x':'PRICE', 'CUSINE_CATEGORY_x':'CUSINE_CATEGORY',
                                'REGION':'REGION', 'CUSINE TYPE_x':'CUSINE TYPE'},inplace=True)
```

In [118]:

```
# Affordable Restaurants with low price and high rating

highrate_aff_df
```

Out[118]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE
0	Cake Centre-The Dessert Maker	150	Desserts	Mohammad Ali Road	Dessert Parlor
1	Curry And Combos Twist	200	North Indian,Chinese	Andheri	Quick Bites
2	Moussestruck	200	Desserts	Near Andheri Station	none
3	Cone Culture	250	European	Kharghar	Casual Dining
4	Smiley Pops	300	Desserts,Ice Cream,Beverages,Sandwich	Andheri	Dessert Parlor
...	...	...	...	...	...
60	Wild Dining Restaurant	1200	North Indian,Continental,Mexican,Chinese	Andheri	Casual Dining
61	Invento	1200	Chinese,Fast Food,North Indian,Italian,Mexican	Lower Parel	Casual Dining
62	Culinary Tales	1200	Chinese,European,Continental,Salad,Italian,Pizza	Veera Desai Area	Casual Dining
63	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	Bandra	Casual Dining
64	The Joker Bistro	1200	North Indian,Chinese,Continental	CBD Belapur	Casual Dining

65 rows × 5 columns

Q11) Find the list of most Reliable Restaurants?

The criteria for most Reliable Restaurants would be:- 1) Low Price 2) High Rated 3) Large No. of Votes

First step will be to find the restaurants with Votes greater than Mean of Votes

In [119]:

```
mean_votes = zomato_df['VOTES'].mean()
mean_votes
```

Out[119]:

```
177.2656679624538
```

In [120]:

```
# Finding list of restaurants that have Votes greater than and equal to  Mean of Vote

mean_rest_df = zomato_df[['NAME', 'PRICE', 'CUSINE_CATEGORY', 'REGION', 'CUSINE TYPE', 'VOTES']]
mean_rest_df = mean_rest_df[mean_rest_df['VOTES'] > 177]
mean_rest_df.sort_values(by='VOTES', inplace=True)
mean_rest_df
```

Out[120]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE	VOTES
4194	Sai Sagar Veg Treat	500	North Indian,South Indian,Chinese,Fast Food,Be...	Kalyan	Casual Dining	178
884	Rajmanya- Seafood family restaurant	800	Maharashtrian,Konkan,Seafood	Vashi	Casual Dining	178
3914	Ice Cafe	500	Fast Food,Ice Cream,Beverages,Pizza	Borivali	Quick Bites	178
7897	Konkan Lajjatdar	500	Seafood,Biryani,Beverages,Chinese,Malwani,Konkan	Andheri	Casual Dining	178
3828	Frozen Delight -The Dessert Cafe	250	Desserts,Ice Cream	Airoli	Dessert Parlor	178
...	...	...	...	...	...	...
8539	Leopold Cafe & Bar	1600	American,Chinese,Mughlai,Italian	Colaba	Casual Dining	7327
1251	Joey's Pizza	800	Pizza	Malad	Quick Bites	7350
5337	Chili's American Grill & Bar	1400	American,Mexican,Burger,Tex-Mex	Powai	Casual Dining	7377
3751	Prithvi Cafe	700	Cafe,Fast Food	Juhu	Café	8000
8897	Candies	700	Cafe,Italian,North Indian,Desserts	Bandra	Café	10217

2345 rows × 6 columns

These are the most reliable, highest rated and affordable restaurants:- We obtain this dataframe by simply taking the intersection of highrate_aff_df & mean_rest_df

This dataframe obtained below shows the restaurants whose:

Cost is below 1250, Rating is above 4.5, Votes are above 177

In [121]:

```
reliable_rest_df = pd.merge(mean_rest_df, highrate_aff_df, how='inner', on=['NAME', 'REGION'])
reliable_rest_df = reliable_rest_df[['NAME', 'PRICE_x', 'CUSINE_CATEGORY_x', 'REGION', 'CUSINE TYPE_x']]
reliable_rest_df.rename(columns={'NAME':'NAME', 'PRICE_x':'PRICE', 'CUSINE_CATEGORY_x':'CUSINE_CATEGORY',
                                'REGION':'REGION', 'CUSINE TYPE_x':'CUSINE TYPE'},inplace=True)
```

In [122]:

```
reliable_rest_df
```

Out[122]:

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE
0	Rajmanya- Seafood family restaurant	800	Maharashtrian,Konkan,Seafood	Vashi	Casual Dining
1	Fresh Food Co.	500	Continental,Healthy Food,Salad,Beverages,Desse...	Santacruz	none
2	Dessertino	300	Desserts,Ice Cream	Kandivali	Dessert Parlor
3	Invento	1200	Chinese,Fast Food,North Indian,Italian,Mexican	Lower Parel	Casual Dining
4	Maharashtra Lunch Home	600	Maharashtrian,Malwani,Konkan,Seafood	Kharghar	Casual Dining
5	Regano's	600	Continental,Fast Food,Italian,Desserts	Malad	Casual Dining
6	Big Bang Cuurry	350	North Indian,Biryani,Rolls	Andheri	none
7	Sandy's Den	1000	Fast Food,Bar Food	Chembur	Casual Dining
8	Angrezi Patiyalaa	1200	North Indian,Finger Food,American,Mexican,Chinese	Andheri	Casual Dining
9	Maezo	1000	Modern Indian	Thane	Casual Dining
10	Tossin Pizza	900	Pizza,Italian,Fast Food	Chembur	Casual Dining
11	Little West Pizza	600	Pizza	Borivali	Quick Bites
12	Ceremonial Kitchen & Co	1000	Seafood,Maharashtrian,North Indian,Chinese	Thane	Casual Dining
13	Moussestruck	200	Desserts	Near Andheri Station	none
14	Daftar Goregaon	750	Pizza,Chinese,North Indian,Beverages	Goregaon	Casual Dining
15	Poetry By Love & Cheesecake	1000	Cafe,Desserts	Juhu	Café
16	Makhan Singh	800	North Indian,Chinese,Biryani	Powai	none
17	The Joker Bistro	1200	North Indian,Chinese,Continental	CBD Belapur	Casual Dining
18	Cone Culture	250	European	Kharghar	Casual Dining
19	Peco Peco	700	Chinese,Seafood,Asian	Powai	none
20	Shuruwat- Veg Food Journey	600	Continental,Tea,South Indian,Fast Food,Pizza,N...	Ghatkopar	Casual Dining
21	Justice Cafe and Dine	800	Cafe,Chinese,Italian,Continental,North Indian,...	Thane	Café
22	Thangabali	1000	Seafood,South Indian,Mangalorean,Andhra,Kerala	Khar	Bar
23	Harsh's Bistro	800	Chinese,Continental	Malad	Casual Dining

	NAME	PRICE	CUSINE_CATEGORY	REGION	CUSINE TYPE
24	Quarter Canteen	1100	North Indian,Seafood,Chinese	Bandra	Casual Dining
25	Culinary Tales	1200	Chinese,European,Continental,Salad,Italian,Pizza	Veera Desai Area	Casual Dining
26	Dum & Curry	700	Mughlai,North Indian,Chinese	Powai	Quick Bites
27	Curry Culture	800	North Indian,Biryani,Chinese,Kebab,Mughlai,Asian	Powai	none
28	Belo Pops	300	Ice Cream,Desserts,Beverages	Andheri	none
29	Tick-eat	800	North Indian,Italian,Chinese,Mexican,Lebanese	Mulund	Casual Dining
30	Coppetto Artisan Gelato	350	Ice Cream,Desserts	Bandra	Dessert Parlor
31	Zaika Restaurant & Party Hall	1000	North Indian,Chinese,Beverages	Bhayandar	Casual Dining
32	Shaollin Temple	1000	Chinese,Thai	CBD Belapur	Casual Dining
33	Spice Republic	1200	Cafe,Continental,Mediterranean,Mexican,Italian...	Borivali	Café
34	Aquafire Restaurant	1100	North Indian,Continental,Chinese,Italian	Vile Parle	Casual Dining
35	Rajdhani	950	Gujarati,Rajasthani,North Indian	Ghatkopar	Casual Dining
36	Wild Dining Restaurant	1200	North Indian,Continental,Mexican,Chinese	Andheri	Casual Dining
37	Family Tree	800	Italian,Mexican,North Indian,Chinese,Salad	Thane	Casual Dining
38	Bombay Salad Co.	900	Salad,Healthy Food,Juices	Bandra	Casual Dining
39	Cafe Monza	1000	Italian,American,Salad,Mexican	Kharghar	Casual Dining
40	Hitchki	1200	Modern Indian,North Indian,Chinese,Momos,Birya...	Bandra	Casual Dining
41	Vedge	1000	Thai,Chinese,North Indian,Mexican,Italian,Asian	Andheri	Casual Dining
42	Joey's Pizza	800	Pizza	Malad	Quick Bites

```
In [ ]:
```