DATA 690 FINAL POJECT NOTEBOOK

FIRST Things First I created three different data frames for three differeent data sets and then printed the heads of the datasets and also used .info() to get more information about the data.

In [1]:
```
import pandas as pd
# Load data from CSV files
business_data = pd.read_csv("yelp_academic_dataset_business.csv")
review_data = pd.read_csv("yelp_academic_dataset_review.csv")
user_data = pd.read_csv("yelp_academic_dataset_user.csv")
# Explore the first few rows of each dataset
print("Business data:")
print(business_data.head())
print("\nReview data:")
print(review_data.head())
print("\nUser data:")
print(user_data.head())
# Get information about each dataset
print("Business data info:")
print(business_data.info())
print("\nReview data info:")
print(review_data.info())
print("\nUser data info:")
print(user_data.info())
```

```
Business data:
                address  attributes attributes.AcceptsInsurance  \
0    1314 44 Avenue NE         NaN                          NaN
1                  NaN         NaN                          NaN
2  1335 rue Beaubien E         NaN                          NaN
3      211 W Monroe St         NaN                          NaN
4  2005 Alyth Place SE         NaN                          NaN

  attributes.AgesAllowed attributes.Alcohol  \
0                    NaN                NaN
1                    NaN               none
2                    NaN      beer_and_wine
3                    NaN                NaN
4                    NaN                NaN

                                attributes.Ambience attributes.BYOB
\
0                                               NaN             NaN
1                                               NaN             NaN
```

```
2  {'romantic': False, 'intimate': False, 'classy...            NaN
3                                                  NaN            NaN
4                                                  NaN            NaN

  attributes.BYOBCorkage attributes.BestNights attributes.BikeParking
...  \
0                    NaN                   NaN                  False
...
1                    NaN                   NaN                  False
...
2                    NaN                   NaN                   True
...
3                    NaN                   NaN                    NaN
...
4                    NaN                   NaN                    NaN
...

  hours.Wednesday is_open   latitude   longitude                   nam
e  \
0       11:0-21:0       1  51.091813 -114.031675  Minhas Micro Brewer
y
1             NaN       0  35.960734 -114.939821   CK'S BBQ & Caterin
g
2       10:0-22:0       0  45.540503  -73.599300         La Bastringu
e
3             NaN       1  33.449999 -112.076979       Geico Insuranc
e
4        8:0-17:0       1  51.035591 -114.027366         Action Engin
e

               neighborhood postal_code review_count stars state
0                       NaN     T2E 6L6           24   4.0    AB
1                       NaN       89002            3   4.5    NV
2  Rosemont-La Petite-Patrie     H2G 1K7            5   4.0    QC
3                       NaN       85003            8   1.5    AZ
4                       NaN     T2H 0N5            4   2.0    AB

[5 rows x 61 columns]

Review data:
              business_id  cool        date  funny               revi
ew_id  \
0  iCQpiavjjPzJ5_3gPD5Ebg     0  2011-02-25    0.0  x7mDIiDB3jEiPGPHO
mDzyw
1  pomGBqfbxcqPv14c3XH-ZQ     0  2012-11-13    0.0  dDl8zu1vWPdKGihJr
wQbpw
2  jtQARsP6P-LbkyjbO1qNGg     1  2014-10-23    1.0  LZp4UX5zK3e-c5ZGS
eo3kA
3  elqbBhBfElMNSrjFqW3now     0  2011-02-25    0.0  Er4NBWCmCD4nM8_p1
GRdow
```

```
4  Ums3gaP2qM3W1XcA5r6SsQ     0  2014-09-05    0.0  jSDu6QEJHbwP2Blom
1PLCA

   stars                                               text  useful
\
0    2.0  The pizza was okay. Not the best I've had. I p...     0.0
1    5.0  I love this place! My fiance And I go here atl...     0.0
2    1.0  Terrible. Dry corn bread. Rib tips were all fa...     3.0
3    2.0  Back in 2005-2007 this place was my FAVORITE t...     2.0
4    5.0  Delicious healthy food. The steak is amazing. ...     0.0

                 user_id
0  msQe1u7Z_XuqjGoqhB0J5g
1  msQe1u7Z_XuqjGoqhB0J5g
2  msQe1u7Z_XuqjGoqhB0J5g
3  msQe1u7Z_XuqjGoqhB0J5g
4  msQe1u7Z_XuqjGoqhB0J5g

User data:
   average_stars  compliment_cool  compliment_cute  compliment_funny
\
0           2.00                0                0                 0
1           5.00                0                0                 0
2           4.00                0                0                 0
3           4.05                0                0                 0
4           3.00                0                0                 0

   compliment_hot  compliment_list  compliment_more  compliment_note
\
0               0                0                0                0
1               0                0                0                0
2               0                0                0                0
3               0                0                0                0
4               0                0                0                0

   compliment_photos  compliment_plain  ...  cool  elite  fans friend
s  funny  \
0                  0                 0  ...     0    NaN     0     Na
N      0
1                  0                 0  ...     0    NaN     0     Na
N      0
2                  0                 0  ...     0    NaN     0     Na
N      0
3                  0                 0  ...     0    NaN     0     Na
N      0
4                  0                 0  ...     0    NaN     0     Na
N      0

       name  review_count useful                 user_id  yelping_sin
ce
```

```
0     Susan             1      0  lzlZwIpuSWXEnNS91wxjHw     2015-09-
28
1  Daipayan             2      0  XvLBr-9smbI0m_a7dXtB7w     2015-09-
05
2      Andy             1      0  QPT4Ud4H5sJVr68yXhoWFw     2016-07-
21
3  Jonathan            19      0  i5YitlHZpf0B3R0s_8NVuw     2014-08-
04
4  Shashank             3      0  s4FoIXE_LSGviTHBe8dmcg     2017-06-
18

[5 rows x 22 columns]
Business data info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188593 entries, 0 to 188592
Data columns (total 61 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   address                               180970 non-null  object
 1   attributes                            0 non-null       float64
 2   attributes.AcceptsInsurance           11671 non-null   object
 3   attributes.AgesAllowed                397 non-null     object
 4   attributes.Alcohol                    47892 non-null   object
 5   attributes.Ambience                   47577 non-null   object
 6   attributes.BYOB                       911 non-null     object
 7   attributes.BYOBCorkage                1409 non-null    object
 8   attributes.BestNights                 6844 non-null    object
 9   attributes.BikeParking                84891 non-null   object
 10  attributes.BusinessAcceptsBitcoin     12674 non-null   object
 11  attributes.BusinessAcceptsCreditCards 140391 non-null  object
 12  attributes.BusinessParking            103424 non-null  object
 13  attributes.ByAppointmentOnly          45423 non-null   object
 14  attributes.Caters                     40038 non-null   object
 15  attributes.CoatCheck                  8531 non-null    object
 16  attributes.Corkage                    657 non-null     object
 17  attributes.DietaryRestrictions        138 non-null     object
 18  attributes.DogsAllowed                13681 non-null   object
 19  attributes.DriveThru                  6754 non-null    object
 20  attributes.GoodForDancing             9162 non-null    object
 21  attributes.GoodForKids                64931 non-null   object
 22  attributes.GoodForMeal                47483 non-null   object
 23  attributes.HairSpecializesIn          1881 non-null    object
 24  attributes.HappyHour                  9285 non-null    object
 25  attributes.HasTV                      47533 non-null   object
 26  attributes.Music                      8807 non-null    object
 27  attributes.NoiseLevel                 43710 non-null   object
 28  attributes.Open24Hours                352 non-null     object
 29  attributes.OutdoorSeating             54181 non-null   object
 30  attributes.RestaurantsAttire          48182 non-null   object
 31  attributes.RestaurantsCounterService  397 non-null     object
```

```
 32  attributes.RestaurantsDelivery         51668 non-null   object
 33  attributes.RestaurantsGoodForGroups    53839 non-null   object
 34  attributes.RestaurantsPriceRange2      107120 non-null  float64
 35  attributes.RestaurantsReservations     51363 non-null   object
 36  attributes.RestaurantsTableService     43325 non-null   object
 37  attributes.RestaurantsTakeOut          61206 non-null   object
 38  attributes.Smoking                     8113 non-null    object
 39  attributes.WheelchairAccessible        52023 non-null   object
 40  attributes.WiFi                        49026 non-null   object
 41  business_id                            188593 non-null  object
 42  categories                             188052 non-null  object
 43  city                                   188583 non-null  object
 44  hours                                  0 non-null       float64
 45  hours.Friday                           141796 non-null  object
 46  hours.Monday                           132761 non-null  object
 47  hours.Saturday                         125376 non-null  object
 48  hours.Sunday                           93387 non-null   object
 49  hours.Thursday                         142359 non-null  object
 50  hours.Tuesday                          140607 non-null  object
 51  hours.Wednesday                        141843 non-null  object
 52  is_open                                188593 non-null  int64
 53  latitude                               188587 non-null  float64
 54  longitude                              188587 non-null  float64
 55  name                                   188593 non-null  object
 56  neighborhood                           68655 non-null   object
 57  postal_code                            187912 non-null  object
 58  review_count                           188593 non-null  int64
 59  stars                                  188593 non-null  float64
 60  state                                  188593 non-null  object
dtypes: float64(6), int64(2), object(53)
memory usage: 87.8+ MB
None

Review data info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5996998 entries, 0 to 5996997
Data columns (total 9 columns):
 #   Column       Dtype
---  ------       -----
 0   business_id  object
 1   cool         int64
 2   date         object
 3   funny        float64
 4   review_id    object
 5   stars        float64
 6   text         object
 7   useful       float64
 8   user_id      object
dtypes: float64(3), int64(1), object(5)
memory usage: 411.8+ MB
```

```
None

User data info:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1518169 entries, 0 to 1518168
Data columns (total 22 columns):
 #   Column              Non-Null Count    Dtype  
---  ------              --------------    -----  
 0   average_stars       1518169 non-null  float64
 1   compliment_cool     1518169 non-null  int64  
 2   compliment_cute     1518169 non-null  int64  
 3   compliment_funny    1518169 non-null  int64  
 4   compliment_hot      1518169 non-null  int64  
 5   compliment_list     1518169 non-null  int64  
 6   compliment_more     1518169 non-null  int64  
 7   compliment_note     1518169 non-null  int64  
 8   compliment_photos   1518169 non-null  int64  
 9   compliment_plain    1518169 non-null  int64  
 10  compliment_profile  1518169 non-null  int64  
 11  compliment_writer   1518169 non-null  int64  
 12  cool                1518169 non-null  int64  
 13  elite               67109 non-null    object 
 14  fans                1518169 non-null  int64  
 15  friends             879891 non-null   object 
 16  funny               1518169 non-null  int64  
 17  name                1517675 non-null  object 
 18  review_count        1518169 non-null  int64  
 19  useful              1518169 non-null  int64  
 20  user_id             1518169 non-null  object 
 21  yelping_since       1518169 non-null  object 
dtypes: float64(1), int64(16), object(5)
memory usage: 254.8+ MB
None
```

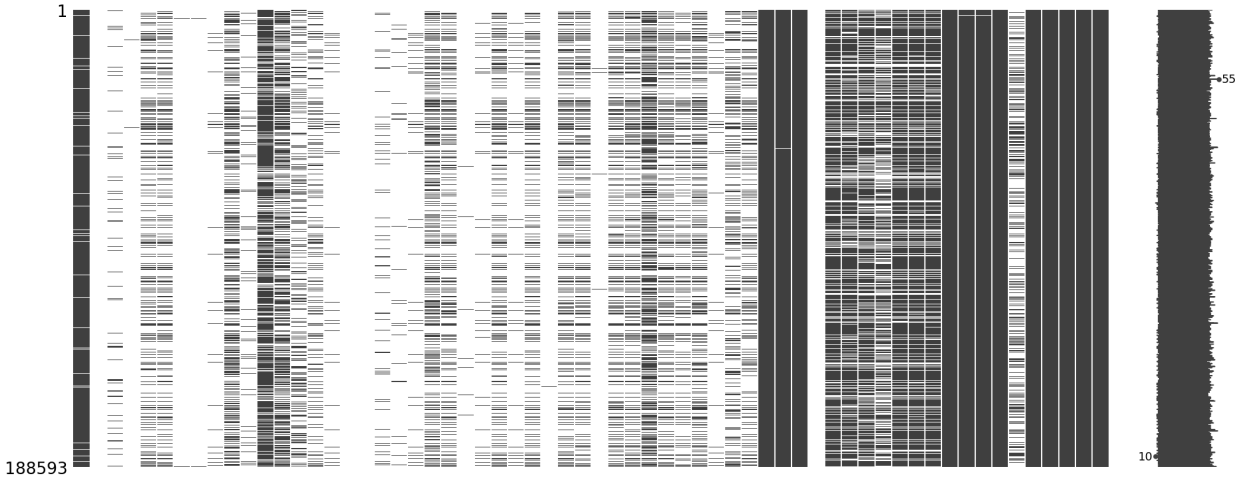
Through the output it is evident that the additional cleaning is required because it contains lot of null values in it.

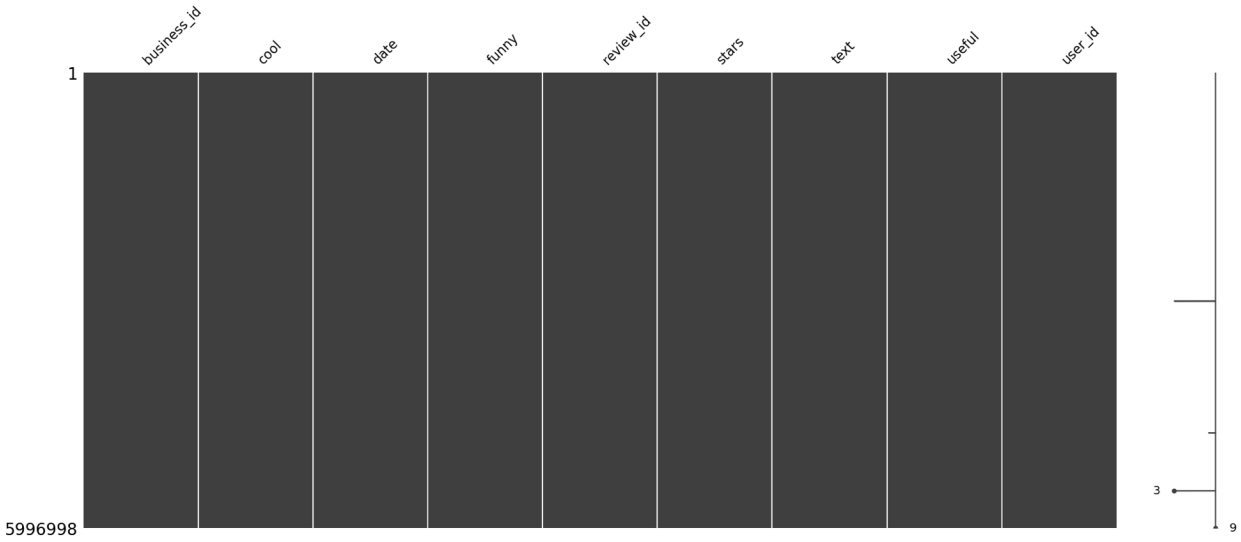
In [6]:
```
!pip install missingno
```

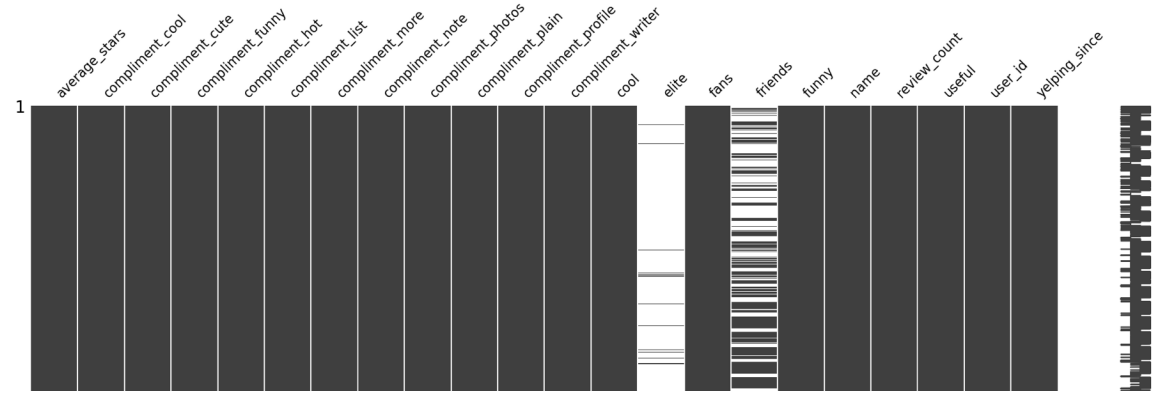
```
Collecting missingno
  Downloading missingno-0.5.2-py3-none-any.whl (8.7 kB)
Requirement already satisfied: numpy in /Applications/anaconda3/lib/p
ython3.11/site-packages (from missingno) (1.24.3)
Requirement already satisfied: matplotlib in /Applications/anaconda3/
lib/python3.11/site-packages (from missingno) (3.7.2)
Requirement already satisfied: scipy in /Applications/anaconda3/lib/p
ython3.11/site-packages (from missingno) (1.11.1)
Requirement already satisfied: seaborn in /Applications/anaconda3/li
b/python3.11/site-packages (from missingno) (0.12.2)
Requirement already satisfied: contourpy>=1.0.1 in /Applications/anac
onda3/lib/python3.11/site-packages (from matplotlib->missingno) (1.0.
5)
Requirement already satisfied: cycler>=0.10 in /Applications/anaconda
3/lib/python3.11/site-packages (from matplotlib->missingno) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in /Applications/ana
conda3/lib/python3.11/site-packages (from matplotlib->missingno) (4.2
5.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /Applications/ana
conda3/lib/python3.11/site-packages (from matplotlib->missingno) (1.
4.4)
Requirement already satisfied: packaging>=20.0 in /Applications/anaco
nda3/lib/python3.11/site-packages (from matplotlib->missingno) (23.1)
Requirement already satisfied: pillow>=6.2.0 in /Applications/anacond
a3/lib/python3.11/site-packages (from matplotlib->missingno) (9.4.0)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in /Application
s/anaconda3/lib/python3.11/site-packages (from matplotlib->missingno)
(3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in /Applications/
anaconda3/lib/python3.11/site-packages (from matplotlib->missingno) (
2.8.2)
Requirement already satisfied: pandas>=0.25 in /Applications/anaconda
3/lib/python3.11/site-packages (from seaborn->missingno) (2.0.3)
Requirement already satisfied: pytz>=2020.1 in /Applications/anaconda
3/lib/python3.11/site-packages (from pandas>=0.25->seaborn->missingn
o) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in /Applications/anacon
da3/lib/python3.11/site-packages (from pandas>=0.25->seaborn->missing
no) (2023.3)
Requirement already satisfied: six>=1.5 in /Applications/anaconda3/li
b/python3.11/site-packages (from python-dateutil>=2.7->matplotlib->mi
ssingno) (1.16.0)
Installing collected packages: missingno
Successfully installed missingno-0.5.2
```

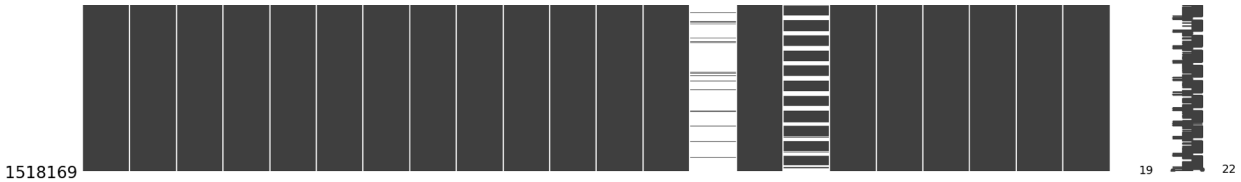
In [7]:
```
import missingno as msno
```

```
import matplotlib.pyplot as plt
# Visualize missing values
msno.matrix(business_data)
plt.show()
msno.matrix(review_data)
plt.show()
msno.matrix(user_data)
plt.show()
```









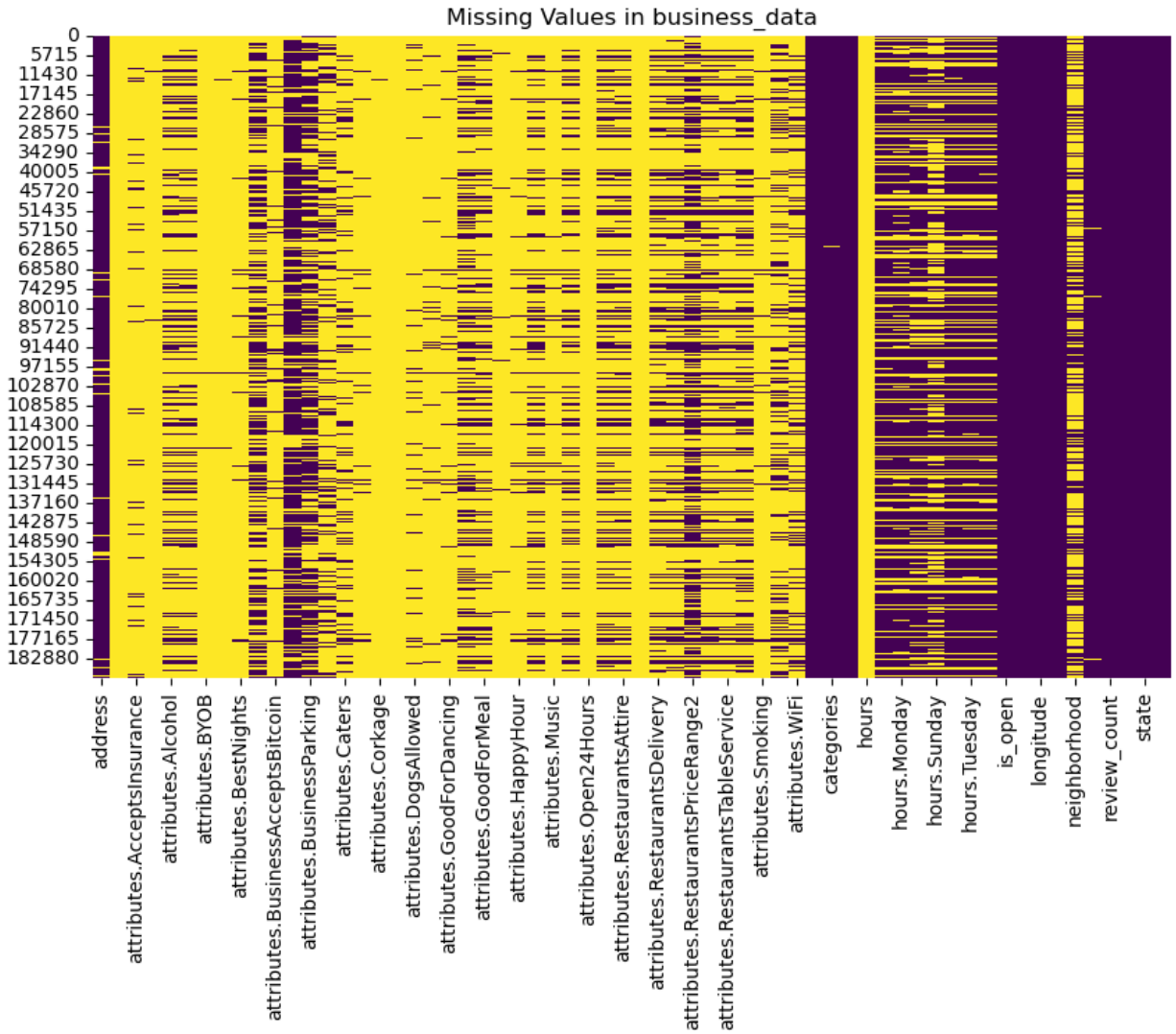
In these i used missingno to vilualise missing values in each data set.

In [8]:
```
# Check for missing values in business_data
missing_business_data = business_data.isnull().sum()
print("Missing values in business_data:")
print(missing_business_data[missing_business_data > 0])
# Check for missing values in review_data
missing_review_data = review_data.isnull().sum()
print("\nMissing values in review_data:")
print(missing_review_data[missing_review_data > 0])
# Check for missing values in user_data
missing_user_data = user_data.isnull().sum()
print("\nMissing values in user_data:")
print(missing_user_data[missing_user_data > 0])
```

```
Missing values in business_data:
address                                 7623
attributes                            188593
attributes.AcceptsInsurance           176922
attributes.AgesAllowed                188196
attributes.Alcohol                    140701
attributes.Ambience                   141016
attributes.BYOB                       187682
attributes.BYOBCorkage                187184
attributes.BestNights                 181749
attributes.BikeParking                103702
attributes.BusinessAcceptsBitcoin     175919
attributes.BusinessAcceptsCreditCards  48202
attributes.BusinessParking             85169
attributes.ByAppointmentOnly          143170
attributes.Caters                     148555
attributes.CoatCheck                  180062
attributes.Corkage                    187936
attributes.DietaryRestrictions        188455
attributes.DogsAllowed                174912
attributes.DriveThru                  181839
attributes.GoodForDancing             179431
attributes.GoodForKids                123662
attributes.GoodForMeal                141110
attributes.HairSpecializesIn          186712
attributes.HappyHour                  179308
attributes.HasTV                      141060
```

```
attributes.Music                          179786
attributes.NoiseLevel                     144883
attributes.Open24Hours                    188241
attributes.OutdoorSeating                 134412
attributes.RestaurantsAttire              140411
attributes.RestaurantsCounterService      188196
attributes.RestaurantsDelivery            136925
attributes.RestaurantsGoodForGroups       134754
attributes.RestaurantsPriceRange2          81473
attributes.RestaurantsReservations        137230
attributes.RestaurantsTableService        145268
attributes.RestaurantsTakeOut             127387
attributes.Smoking                        180480
attributes.WheelchairAccessible           136570
attributes.WiFi                           139567
categories                                   541
city                                          10
hours                                     188593
hours.Friday                               46797
hours.Monday                               55832
hours.Saturday                             63217
hours.Sunday                               95206
hours.Thursday                             46234
hours.Tuesday                              47986
hours.Wednesday                            46750
latitude                                       6
longitude                                      6
neighborhood                              119938
postal_code                                  681
dtype: int64

Missing values in review_data:
funny        2
review_id    2
stars        2
text         3
useful       4
user_id      4
dtype: int64

Missing values in user_data:
elite      1451060
friends     638278
name           494
dtype: int64
```

In [9]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Visualize missing values in business_data
plt.figure(figsize=(10, 6))
sns.heatmap(business_data.isnull(), cbar=False, cmap='viridis')
plt.title('Missing Values in business_data')
plt.show()
```



In above two cells as the part of preprocessing the data i reviwed each attribute in data and using seaborn i visuallu plotted it

STEP 2 :: CLEANING

In [13]:
```
# Identify numerical and categorical columns
numerical_cols = business_data.select_dtypes(include=['float64', 'int6
categorical_cols = business_data.select_dtypes(include=['object']).col
print("Numerical Columns:")
print(numerical_cols)
print("\nCategorical Columns:")
print(categorical_cols)
# Fill missing values for numerical columns with the mean
business_data[numerical_cols] = business_data[numerical_cols].fillna(b
# Fill missing values for categorical columns with the mode
business_data[categorical_cols] = business_data[categorical_cols].fill
# Check if there are any remaining missing values
remaining_missing = business_data.isnull().sum()
print("\nRemaining Missing Values:")
print(remaining_missing[remaining_missing > 0])
```

```
Numerical Columns:
Index(['attributes', 'attributes.RestaurantsPriceRange2', 'hours', 'i
s_open',
       'latitude', 'longitude', 'review_count', 'stars_x', 'stars_
y'],
      dtype='object')

Categorical Columns:
Index(['address', 'attributes.AgesAllowed', 'attributes.Alcohol',
       'attributes.Ambience', 'attributes.BYOBCorkage',
       'attributes.BestNights', 'attributes.BusinessParking',
       'attributes.DietaryRestrictions', 'attributes.GoodForMeal',
       'attributes.HairSpecializesIn', 'attributes.Music',
       'attributes.NoiseLevel', 'attributes.RestaurantsAttire',
       'attributes.Smoking', 'attributes.WiFi', 'business_id', 'categ
ories',
       'city', 'hours.Friday', 'hours.Monday', 'hours.Saturday',
       'hours.Sunday', 'hours.Thursday', 'hours.Tuesday', 'hours.Wedn
esday',
       'name', 'neighborhood', 'postal_code', 'state'],
      dtype='object')

Remaining Missing Values:
attributes    188593
hours         188593
dtype: int64
```

AS data data contain lot of missing values i did following steps:

*filled missing values for numerical columns with mean. *filled missing values for categorical columns with mode. *and checked are there any missing values.

In [14]:
```
# Drop columns with a high number of missing values
business_data = business_data.drop(['attributes', 'hours'], axis=1)
```

here i dropped the high number of missing values column as it is also not required for future analysis.

In [17]:
```
print(business_data.head())
```

```
                  address  attributes.AcceptsInsurance attributes.Age
sAllowed  \
0       1314 44 Avenue NE                         True
21plus
1  5757 Wayne Newton Blvd                         True
21plus
2     1335 rue Beaubien E                         True
21plus
3         211 W Monroe St                         True
21plus
4     2005 Alyth Place SE                         True
21plus

  attributes.Alcohol                                 attributes.Ambien
ce  \
0               none  {'romantic': False, 'intimate': False, 'class
y...
1               none  {'romantic': False, 'intimate': False, 'class
y...
2      beer_and_wine  {'romantic': False, 'intimate': False, 'class
y...
3               none  {'romantic': False, 'intimate': False, 'class
y...
4               none  {'romantic': False, 'intimate': False, 'class
y...

   attributes.BYOB attributes.BYOBCorkage  \
0            False                    no
1            False                    no
2            False                    no
3            False                    no
```

```
4            False                      no

                                attributes.BestNights  attributes.Bike
Parking  \
0  {'monday': False, 'tuesday': False, 'friday': ...
False
1  {'monday': False, 'tuesday': False, 'friday': ...
False
2  {'monday': False, 'tuesday': False, 'friday': ...
True
3  {'monday': False, 'tuesday': False, 'friday': ...
True
4  {'monday': False, 'tuesday': False, 'friday': ...
True

   attributes.BusinessAcceptsBitcoin  ...  is_open   latitude   longi
tude  \
0                              False  ...        1  51.091813 -114.03
1675
1                              False  ...        0  35.960734 -114.93
9821
2                              False  ...        0  45.540503  -73.59
9300
3                              False  ...        1  33.449999 -112.07
6979
4                              False  ...        1  51.035591 -114.02
7366

                  name               neighborhood  postal_code revie
w_count  \
0  Minhas Micro Brewery                   Westside      T2E 6L6
24
1   CK'S BBQ & Catering                   Westside        89002
3
2         La Bastringue  Rosemont-La Petite-Patrie      H2G 1K7
5
3       Geico Insurance                   Westside        85003
8
4         Action Engine                   Westside      T2H 0N5
4

   stars_x  state  stars_y
0      4.0     AB      4.0
1      4.5     NV      4.5
2      4.0     QC      4.0
3      1.5     AZ      1.5
4      2.0     AB      2.0

[5 rows x 60 columns]
```

In [16]:
```
business_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 188593 entries, 0 to 188592
Data columns (total 60 columns):
 #   Column                                Non-Null Count   Dtype  
---  ------                                --------------   -----  
 0   address                               188593 non-null  object 
 1   attributes.AcceptsInsurance           188593 non-null  bool   
 2   attributes.AgesAllowed                188593 non-null  object 
 3   attributes.Alcohol                    188593 non-null  object 
 4   attributes.Ambience                   188593 non-null  object 
 5   attributes.BYOB                       188593 non-null  bool   
 6   attributes.BYOBCorkage                188593 non-null  object 
 7   attributes.BestNights                 188593 non-null  object 
 8   attributes.BikeParking                188593 non-null  bool   
 9   attributes.BusinessAcceptsBitcoin     188593 non-null  bool   
 10  attributes.BusinessAcceptsCreditCards 188593 non-null  bool   
 11  attributes.BusinessParking            188593 non-null  object 
 12  attributes.ByAppointmentOnly          188593 non-null  bool   
 13  attributes.Caters                     188593 non-null  bool   
 14  attributes.CoatCheck                  188593 non-null  bool   
 15  attributes.Corkage                    188593 non-null  bool   
 16  attributes.DietaryRestrictions        188593 non-null  object 
 17  attributes.DogsAllowed                188593 non-null  bool   
 18  attributes.DriveThru                  188593 non-null  bool   
 19  attributes.GoodForDancing             188593 non-null  bool   
 20  attributes.GoodForKids                188593 non-null  bool   
 21  attributes.GoodForMeal                188593 non-null  object 
 22  attributes.HairSpecializesIn          188593 non-null  object 
 23  attributes.HappyHour                  188593 non-null  bool   
 24  attributes.HasTV                      188593 non-null  bool   
 25  attributes.Music                      188593 non-null  object 
 26  attributes.NoiseLevel                 188593 non-null  object 
 27  attributes.Open24Hours                188593 non-null  bool   
 28  attributes.OutdoorSeating             188593 non-null  bool   
 29  attributes.RestaurantsAttire          188593 non-null  object 
 30  attributes.RestaurantsCounterService  188593 non-null  bool   
 31  attributes.RestaurantsDelivery        188593 non-null  bool   
 32  attributes.RestaurantsGoodForGroups   188593 non-null  bool   
 33  attributes.RestaurantsPriceRange2     188593 non-null  float64
 34  attributes.RestaurantsReservations    188593 non-null  bool   
 35  attributes.RestaurantsTableService    188593 non-null  bool   
 36  attributes.RestaurantsTakeOut         188593 non-null  bool   
 37  attributes.Smoking                    188593 non-null  object 
 38  attributes.WheelchairAccessible       188593 non-null  bool   
 39  attributes.WiFi                       188593 non-null  object 
 40  business_id                           188593 non-null  object 
 41  categories                            188593 non-null  object 
 42  city                                  188593 non-null  object 
 43  hours.Friday                          188593 non-null  object 
```

```
 44  hours.Monday                                188593 non-null  object
 45  hours.Saturday                              188593 non-null  object
 46  hours.Sunday                                188593 non-null  object
 47  hours.Thursday                              188593 non-null  object
 48  hours.Tuesday                               188593 non-null  object
 49  hours.Wednesday                             188593 non-null  object
 50  is_open                                     188593 non-null  int64
 51  latitude                                    188593 non-null  float64
 52  longitude                                   188593 non-null  float64
 53  name                                        188593 non-null  object
 54  neighborhood                                188593 non-null  object
 55  postal_code                                 188593 non-null  object
 56  review_count                                188593 non-null  int64
 57  stars_x                                     188593 non-null  float64
 58  state                                       188593 non-null  object
 59  stars_y                                     188593 non-null  float64
dtypes: bool(24), float64(5), int64(2), object(29)
memory usage: 56.1+ MB
```

remove duplicates in each dataframe

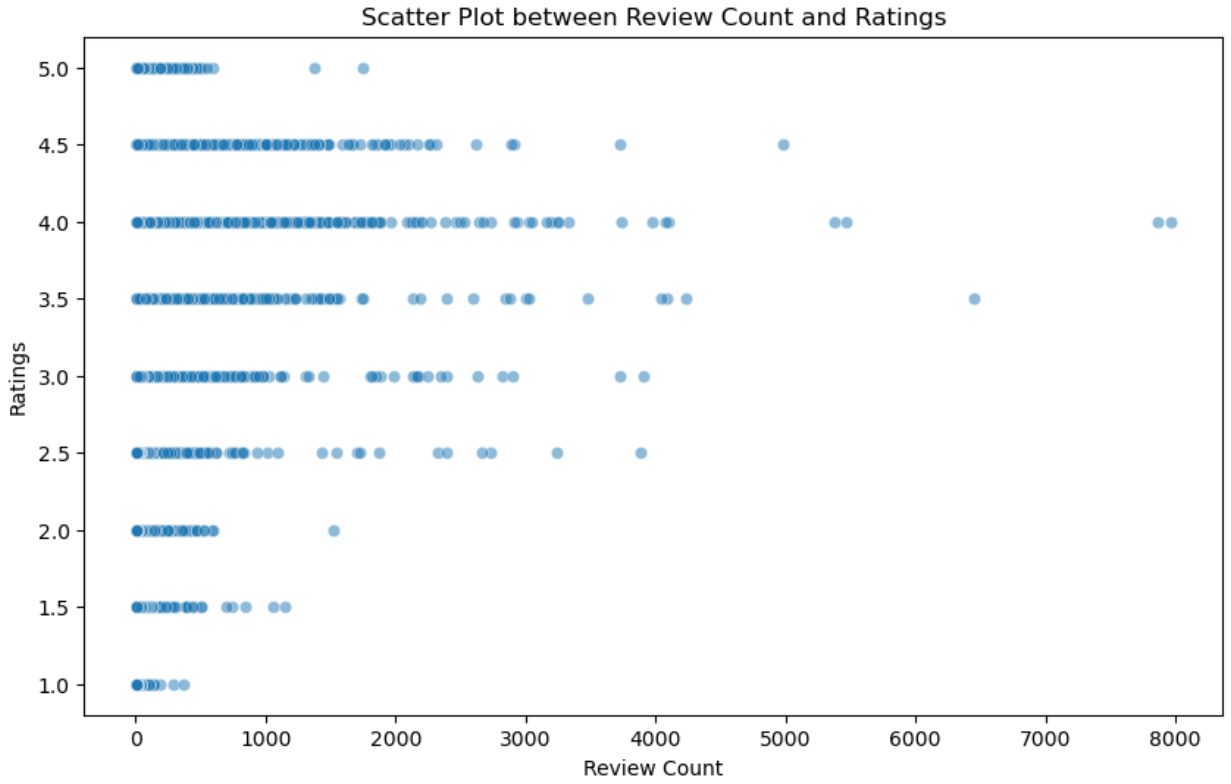
In [18]:
```
# Remove duplicates in each DataFrame
business_data = business_data.drop_duplicates()
review_data = review_data.drop_duplicates()
user_data = user_data.drop_duplicates()
```

ALL SET NOW THE DATASETS ARE PREPROSSED AND READY FOR FUTURE ANALYSIS:

In [ ]:
```

```

Q1 Do businesses with more reviews tend to have higher ratings?

In [25]:
```
import seaborn as sns
import matplotlib.pyplot as plt
# Scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='review_count', y='stars_x', data=business_data, alp
plt.title('Scatter Plot between Review Count and Ratings')
plt.xlabel('Review Count')
plt.ylabel('Ratings')
plt.show()
# Calculate correlation coefficient
correlation_coefficient = business_data['review_count'].corr(business_
print(f'Correlation Coefficient: {correlation_coefficient}')
```



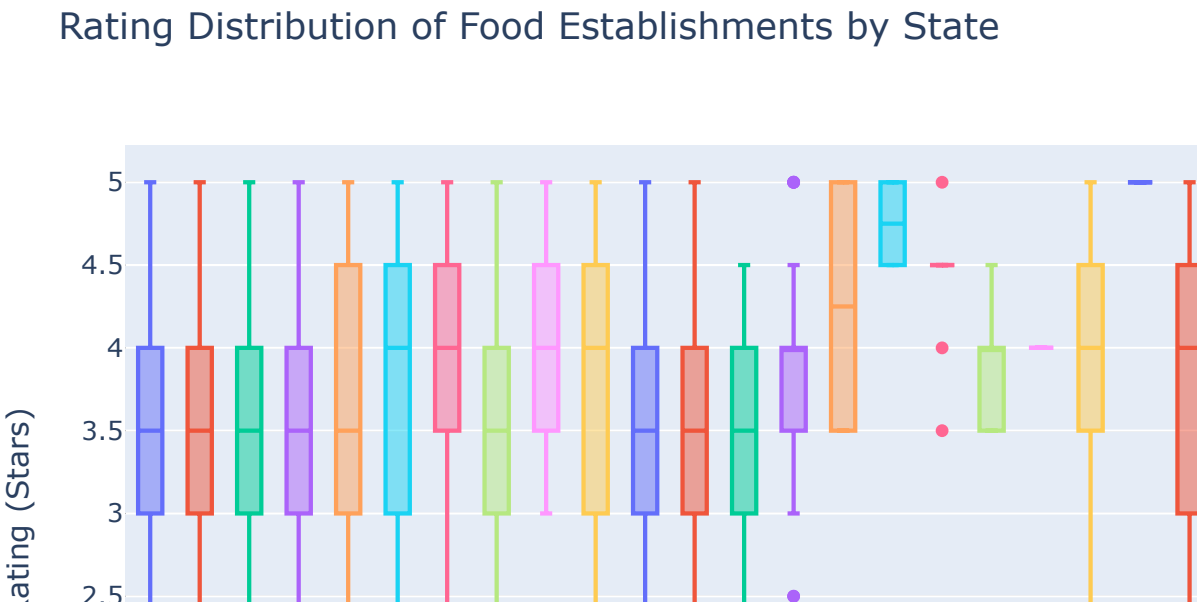

```
Correlation Coefficient: 0.032413313301725755
```

Through these visualiztion we could say that reviews and ratings are not corelated to each other as the correlation coefficient is positive it says that there is no relationship between review and ratings.

There is a dense formation of clusters which says the poor relationship between review and ratings.

Q2 Is there a difference in the rating distribution (stars) of food establishmentsby state?

In [102]:
```
import plotly.express as px
food_establishments = business_data[business_data['categories'].str.co
# Create an interactive box plot
fig = px.box(food_establishments, x='state', y='stars_x', title='Ratin
             labels={'stars_x': 'Rating (Stars)', 'state': 'State'}, c
# Customize the layout
fig.update_layout(xaxis=dict(title='State'), yaxis=dict(title='Rating
# Show the interactive plot
fig.show()
```

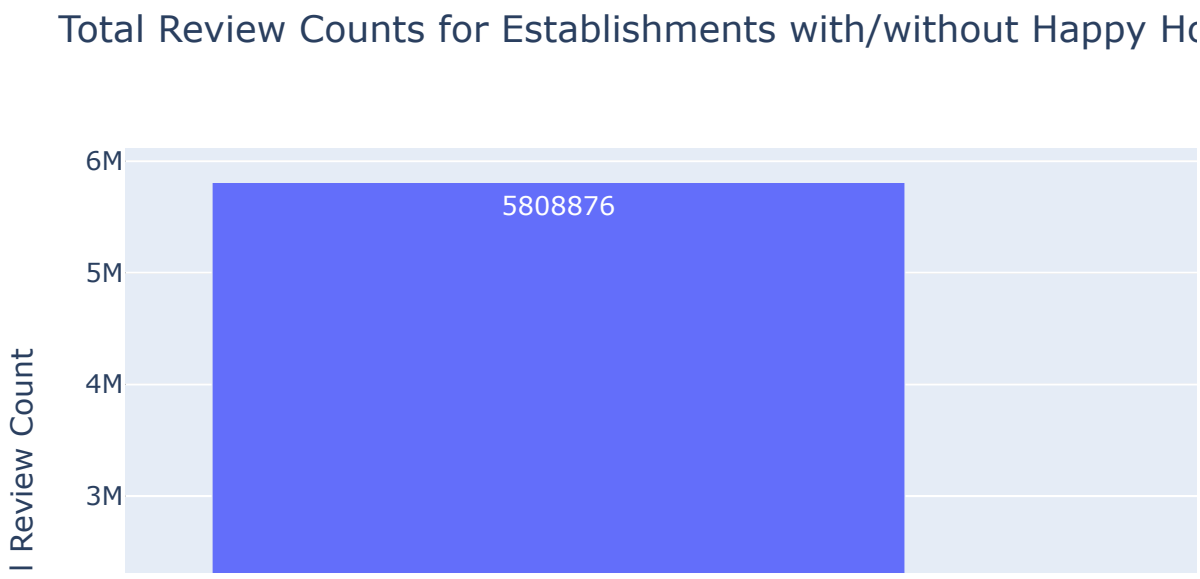


The viz outcome says there is no difference in the rating distribution of food establishments by states each states includes high and lower ratings then above box plot also says the same so we can conclude that there is no difference in rating for each state.

Q3 Investigate differences in food establishments with/without a Happy Hourusing an appropriate visualization.

In [101]:

```
business_data['has_happy_hour'] = business_data['attributes.HappyHour'
# Filter data for establishments with and without Happy Hour
with_happy_hour = business_data[business_data['has_happy_hour'] == 'tr
without_happy_hour = business_data[business_data['has_happy_hour'] ==
# Create an interactive bar plot using Plotly Express
fig = px.bar(x=['With Happy Hour', 'Without Happy Hour'],
             y=[with_happy_hour['review_count'].sum(), without_happy_h
             labels={'x': 'Happy Hour', 'y': 'Total Review Count'},
             title='Total Review Counts for Establishments with/withou
             text=[with_happy_hour['review_count'].sum(), without_happ
             height=500)
# Show the plot
fig.show()
```



There is a difference in total review counts for establishments with/without happy hour there is a huge difference the bar plot says that the the food establishments with happyhour has the more reviews compared to food establishments without happyhour.

Q4 Suppose you work at Yelp. You have been tasked with a new initiative tocreate a new award for the highest performing food establishmentsrepresented on the Yelp platform. The team at Yelp has determined that anyfood establishment that has both the highest stars rating AND the largesttotal number of reviews (review_count) in their city deserves the Best LocalFood Establishment Award. You have been tasked with writing Python codethat can determine which food establishments in a dataset deserve this newprestigious award.

In [100]:
```
# Filter rows with food establishments
food_establishments = business_data[business_data['categories'].str.co
# Group by city and find the establishment with the highest stars rati
best_food_establishments = (
    food_establishments
    .groupby('city')
    .apply(lambda group: group.nlargest(1, 'review_count').nlargest(1,
    .reset_index(drop=True)
)
# Display the establishments deserving the Best Local Food Establishme
print(best_food_establishments[['name', 'city', 'stars_x', 'review_cou
```

```
                                name                        city  sta
rs_x  \
0                          McDonald's                   AGINCOURT
2.0
1                          McDonald's                   Agincourt
3.0
2                       Cupz N' Crepes                   Ahwatukee
4.0
3                    Abe's Restaurant                     Airdrie
4.0
4    The Keg Steakhouse + Bar - Ajax                        Ajax
4.5
..                                ...                         ...
...
597                     Yummy Market  York Regional Municipality
```

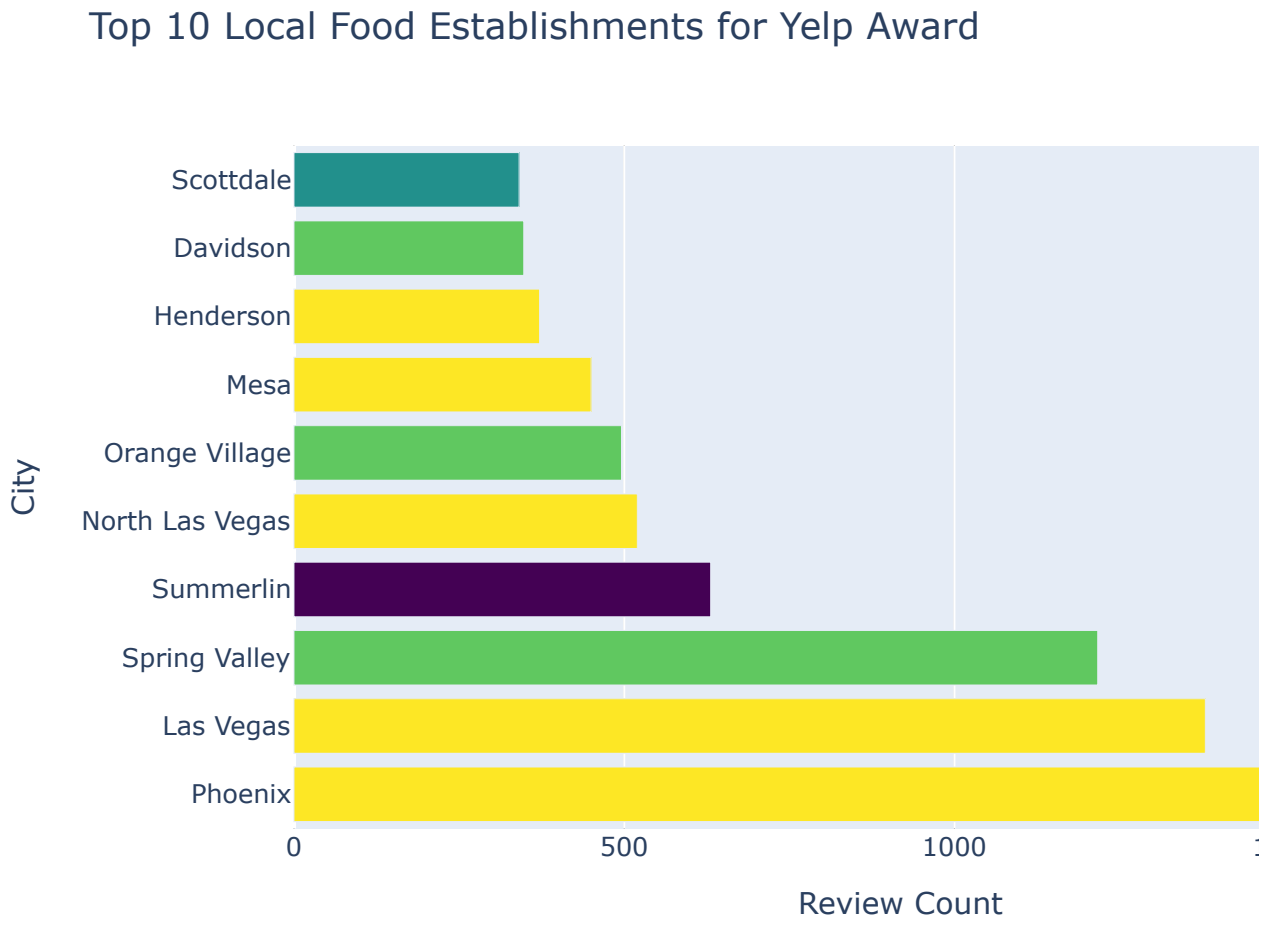
```
3.5
598             Mighty Moo Ice Cream                    Youngtown
5.0
599            Best Grocery Delivery                      clinton
5.0
600              Gibbs Butcher Block            columbia station
4.0
601                      Liquor Fort                    las vegas
3.5

     review_count
0               7
1               4
2             283
3              35
4              45
..            ...
597            43
598           163
599            16
600            33
601            11

[602 rows x 4 columns]
```

In this i first filtered the rows with food establishments and then group by city and find the food establishments with reviews abd ratings and then displays the establishmnets deserving the best local food establishment award

In [99]:

```
food_establishments = business_data[business_data['categories'].str.co
# Grouping by city and get the maximum stars for each city
max_stars_per_city = food_establishments.groupby('city')['stars_x'].ma
food_establishments_with_max_stars = pd.merge(food_establishments, max

# Get the top food establishments with the highest review count in eac
top_food_establishments = food_establishments_with_max_stars.sort_valu
top_food_establishments = top_food_establishments.drop_duplicates(subs

# Select the top 10 food establishments based on review count
top_10_food_establishments = top_food_establishments.nlargest(10, 'rev

# Plotting using Plotly Express
fig = px.bar(top_10_food_establishments, x='review_count', y='city', c
             labels={'review_count': 'Review Count', 'city': 'City'},
             title='Top 10 Local Food Establishments for Yelp Award',
             color_continuous_scale='Viridis')
```

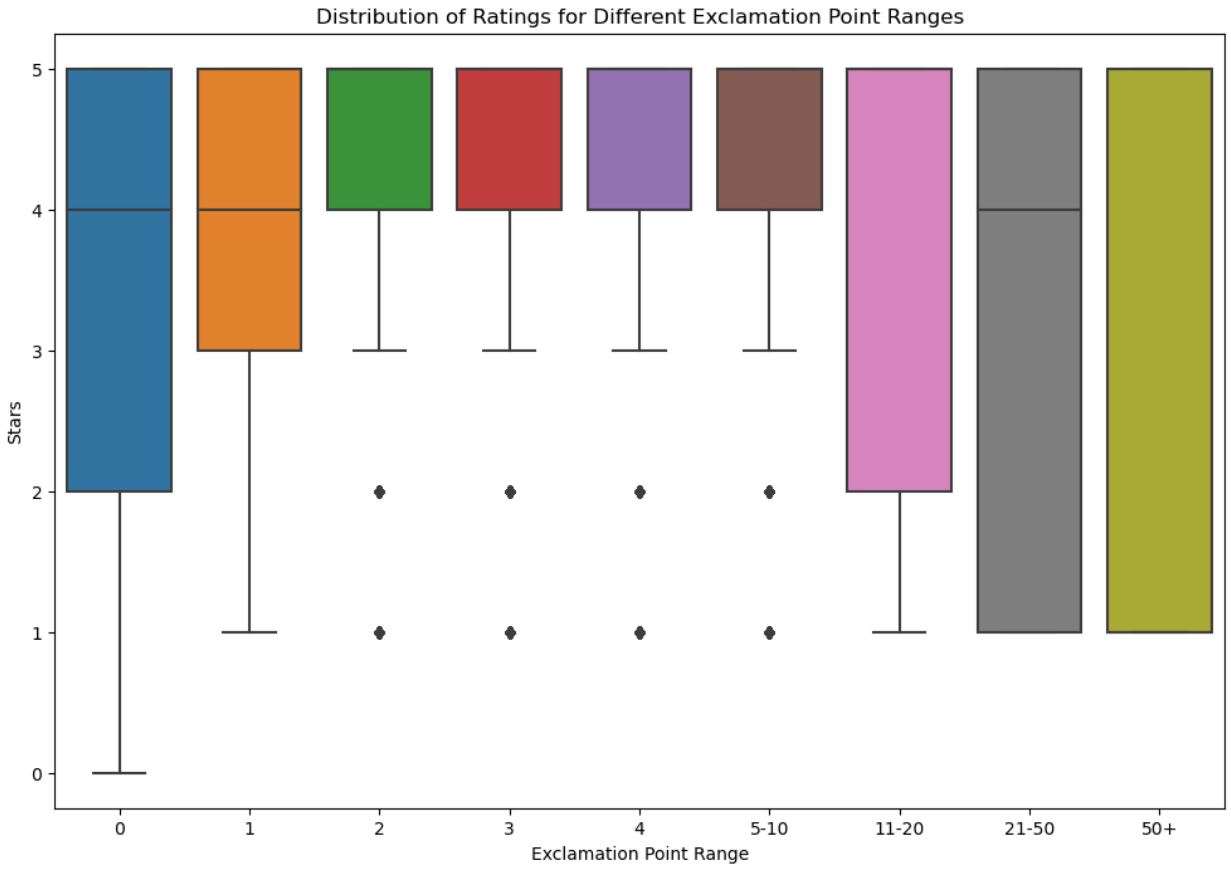
```
# Customize the layout
fig.update_layout(xaxis=dict(title='Review Count'), yaxis=dict(title='
                  legend=dict(title='Stars Rating'), height=500, width

# Show the interactive plot
fig.show()
```



In this i just plotted the top ten local food establishments which eligble for yelp award.

Q5B Do reviews with exclamation points seem to be either very highlyrated or very low rated? Determine the stars distribution as a functionof the number of exclamation points used in the review. Draw asuitable plot.

In [98]:
```
# Check for NaN or float values in the 'text' column and replace them
review_data['text'] = review_data['text'].fillna('')

# Count exclamation points in each review
review_data['exclamation_count'] = review_data['text'].apply(lambda x:

# Define ranges for exclamation points
bins = [0, 1, 2, 3, 4, 5, 10, 20, 50, float('inf')]
labels = ['0', '1', '2', '3', '4', '5-10', '11-20', '21-50', '50+']

# Create a new column for exclamation point ranges
review_data['exclamation_range'] = pd.cut(review_data['exclamation_cou

# Visualize the distribution of ratings for different ranges of exclam
plt.figure(figsize=(12, 8))
sns.boxplot(x='exclamation_range', y='stars', data=review_data, order=
plt.title('Distribution of Ratings for Different Exclamation Point Ran
plt.xlabel('Exclamation Point Range')
plt.ylabel('Stars')
plt.show()
```



Based on the observed trends, yes, there seems to be a tendency for reviews with exclamation points to be either very highly rated or very low rated. The use of exclamation points appears to be associated with expressing strong emotions, which can manifest as either positive or negative reviews depending on the context.

In [ ]: