In [2]:
```
#Import dependencies

import pandas as pd
import missingno as msno
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sqlalchemy import create_engine
```

In [3]:
```
#Read the file listings_details.csv
#The file has 2387 rows and 106 columns


listing_detail_path = "static/data/listings_details.csv"

listing_detail_df = pd.read_csv(listing_detail_path)
listing_detail_df.head()
```

Out[3]:

	id	listing_url	scrape_id	last_scraped	name
0	38585	https://www.airbnb.com/rooms/38585	20200529020107	2020-06-04	Charming Victorian home - twin beds + breakfast
1	80905	https://www.airbnb.com/rooms/80905	20200529020107	2020-06-04	French Chic Loft
2	108061	https://www.airbnb.com/rooms/108061	20200529020107	2020-06-04	Walk to stores/parks/downtown. Fenced yard/Pet...
3	155305	https://www.airbnb.com/rooms/155305	20200529020107	2020-05-29	Cottage! BonPaul + Sharky's Hostel
4	156805	https://www.airbnb.com/rooms/156805	20200529020107	2020-05-29	Private Room "Ader" at BPS Hostel

5 rows × 106 columns

In [4]:
```
# Drop unneeded columns

listing_detail_df_new = listing_detail_df.drop(['scrape_id','listing_url','last_
                         'host_picture_url','host_neighbourhood','requires_licen
                       'is_business_travel_ready','require_guest_profile_picture
                        'neighborhood_overview','notes','transit','access','inte
                        'host_identity_verified','is_location_exact','weekly_pri
                         'guests_included','extra_people','square_feet','calcula
                         'calculated_host_listings_count_private_rooms','calcula
                        'neighbourhood','accommodates','bathrooms','bedrooms','b
```

In [5]:
```
listing_detail_df_new.head()
```

Out[5]:

	id	name	summary	space	description	host_id	host_name	host_
0	38585	Charming Victorian home - twin beds + breakfast	Per the City Council of Asheville. Due to the ...	Charming room with 2 twin size beds, furnished...	Per the City Council of Asheville. Due to the ...	165529	Evelyne	2010-
1	80905	French Chic Loft	Let yourself melt into the delectable décor of...	Have you ever gazed at dreamy photos in a maga...	Let yourself melt into the delectable décor of...	427027	Celeste	2011-
2	108061	Walk to stores/parks/downtown. Fenced yard/Pet...	Walk to town in ten minutes! Monthly rental in...	True Asheville...artist styled apartment with ...	Walk to town in ten minutes! Monthly rental in...	320564	Lisa	2010-
3	155305	Cottage! BonPaul + Sharky's Hostel	NaN	Private cottage located behind the main house ...	Private cottage located behind the main house ...	746673	BonPaul	2011-
4	156805	Private Room "Ader" at BPS Hostel	NaN	Private Rooms at Bon Paul and Sharky's Hostel....	Private Rooms at Bon Paul and Sharky's Hostel....	746673	BonPaul	2011-

5 rows × 59 columns

In [6]:
```
#Analyzing the file for null values
# listing_detail_df_new.isna().sum()

listing_detail_df_new.info()
```

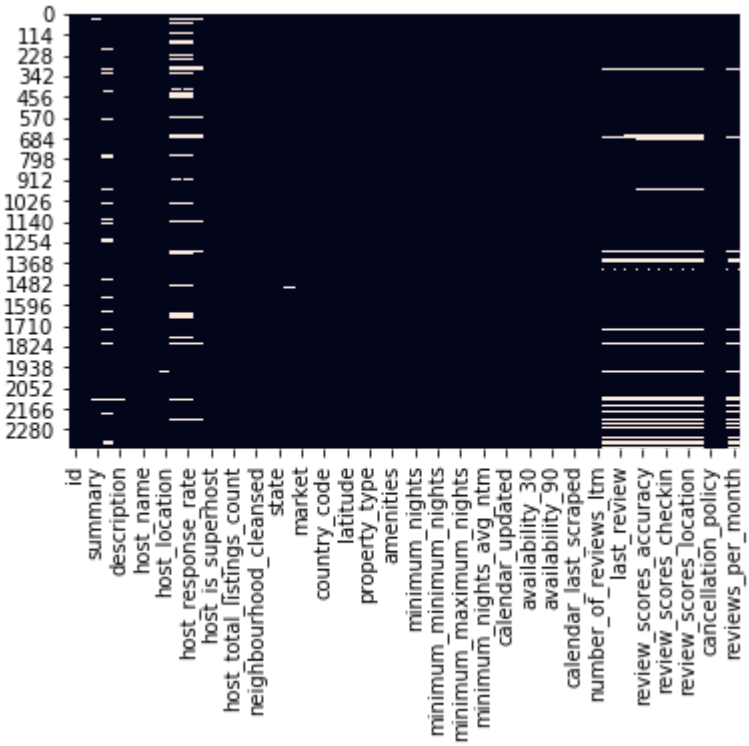
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2387 entries, 0 to 2386
Data columns (total 59 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              2387 non-null   int64  
 1   name                            2387 non-null   object 
 2   summary                         2373 non-null   object 
 3   space                           2108 non-null   object 
 4   description                     2384 non-null   object 
 5   host_id                         2387 non-null   int64  
 6   host_name                       2387 non-null   object 
 7   host_since                      2387 non-null   object 
 8   host_location                   2386 non-null   object 
 9   host_response_time              2004 non-null   object 
 10  host_response_rate              2004 non-null   object 
 11  host_acceptance_rate            2244 non-null   object 
 12  host_is_superhost               2387 non-null   object 
 13  host_listings_count             2387 non-null   int64  
 14  host_total_listings_count       2387 non-null   int64  
 15  street                          2387 non-null   object 
 16  neighbourhood_cleansed          2387 non-null   int64  
 17  city                            2387 non-null   object 
 18  state                           2387 non-null   object 
 19  zipcode                         2365 non-null   float64
 20  market                          2382 non-null   object 
 21  smart_location                  2387 non-null   object 
 22  country_code                    2387 non-null   object 
 23  country                         2387 non-null   object 
 24  latitude                        2387 non-null   float64
 25  longitude                       2387 non-null   float64
 26  property_type                   2387 non-null   object 
 27  room_type                       2387 non-null   object 
 28  amenities                       2387 non-null   object 
 29  price                           2387 non-null   object 
 30  minimum_nights                  2387 non-null   int64  
 31  maximum_nights                  2387 non-null   int64  
 32  minimum_minimum_nights          2387 non-null   int64  
 33  maximum_minimum_nights          2387 non-null   int64  
 34  minimum_maximum_nights          2387 non-null   int64  
 35  maximum_maximum_nights          2387 non-null   int64  
 36  minimum_nights_avg_ntm          2387 non-null   float64
 37  maximum_nights_avg_ntm          2387 non-null   float64
 38  calendar_updated                2387 non-null   object 
 39  has_availability                2387 non-null   object 
 40  availability_30                 2387 non-null   int64  
 41  availability_60                 2387 non-null   int64  
 42  availability_90                 2387 non-null   int64  
 43  availability_365                2387 non-null   int64  
 44  calendar_last_scraped           2387 non-null   object 
 45  number_of_reviews               2387 non-null   int64  
 46  number_of_reviews_ltm           2387 non-null   int64  
```

```
 47  first_review                      2178 non-null   object
 48  last_review                       2178 non-null   object
 49  review_scores_rating              2171 non-null   float64
 50  review_scores_accuracy            2160 non-null   float64
 51  review_scores_cleanliness         2160 non-null   float64
 52  review_scores_checkin             2160 non-null   float64
 53  review_scores_communication       2160 non-null   float64
 54  review_scores_location            2160 non-null   float64
 55  review_scores_value               2160 non-null   float64
 56  cancellation_policy               2387 non-null   object
 57  calculated_host_listings_count    2387 non-null   int64
 58  reviews_per_month                 2178 non-null   float64
dtypes: float64(13), int64(18), object(28)
memory usage: 1.1+ MB
```

In [7]:

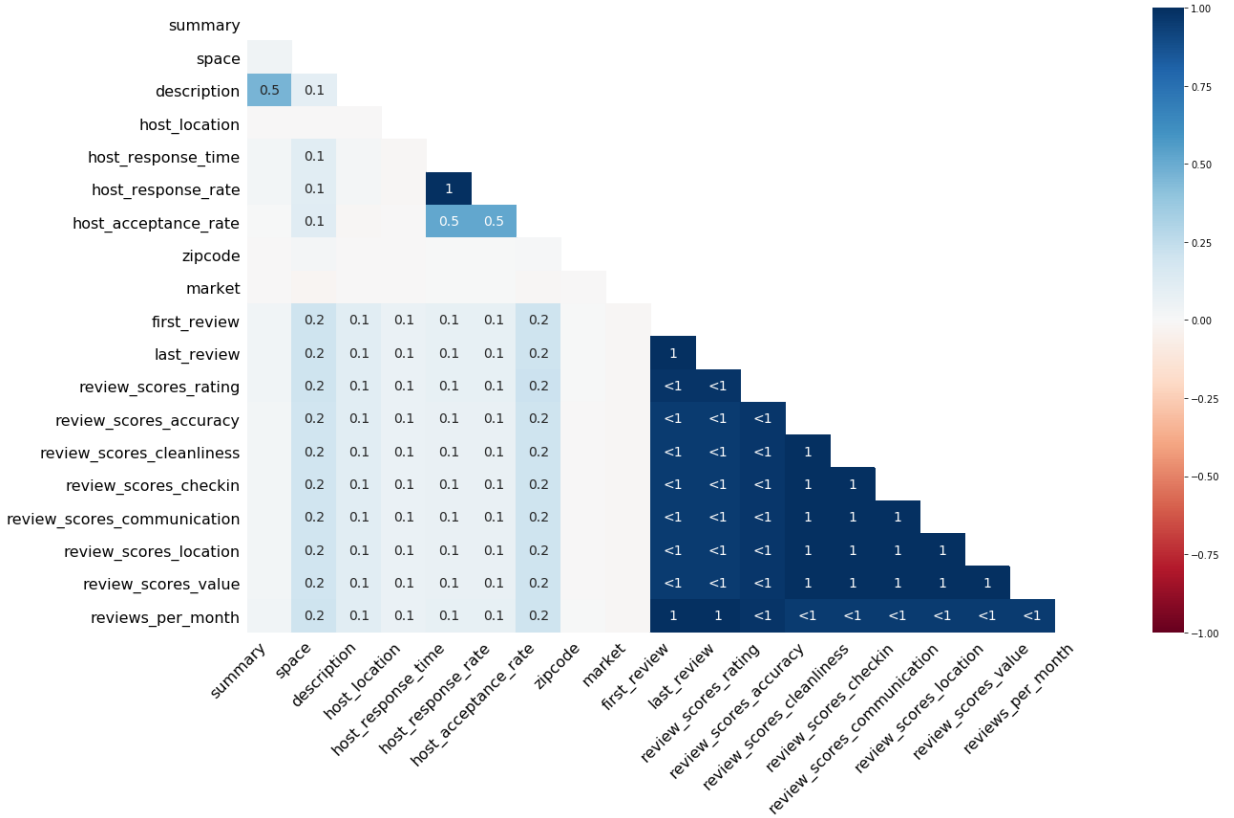
```
sns.heatmap(listing_detail_df_new.isnull(), cbar=False)
```

Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x15c08b55808>



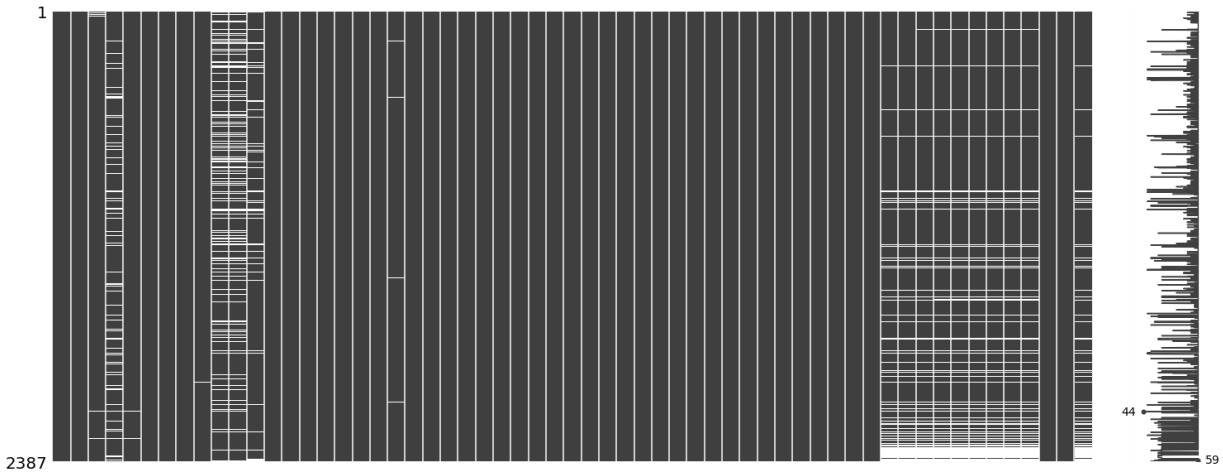
In [11]:
```
msno.heatmap(listing_detail_df_new)
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x15c0a24b888>



In [12]:
```
msno.matrix(listing_detail_df_new)
```

Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15c0a907cc8>
```



In [13]:
```
# listing_detail_df_new.dtypes
# Not dropping any null values after inspecting
# Replacing all null values with 0

listing_detail_df_new.replace(to_replace = np.nan, value =0,inplace=True)

listing_detail_df_new.head()
```

Out[13]:

	id	name	summary	space	description	host_id	host_name	host_
0	38585	Charming Victorian home - twin beds + breakfast	Per the City Council of Asheville. Due to the ...	Charming room with 2 twin size beds, furnished...	Per the City Council of Asheville. Due to the ...	165529	Evelyne	2010-
1	80905	French Chic Loft	Let yourself melt into the delectable décor of...	Have you ever gazed at dreamy photos in a maga...	Let yourself melt into the delectable décor of...	427027	Celeste	2011-
2	108061	Walk to stores/parks/downtown. Fenced yard/Pet...	Walk to town in ten minutes! Monthly rental in...	True Asheville...artist styled apartment with ...	Walk to town in ten minutes! Monthly rental in...	320564	Lisa	2010-
3	155305	Cottage! BonPaul + Sharky's Hostel	0	Private cottage located behind the main house ...	Private cottage located behind the main house ...	746673	BonPaul	2011-
4	156805	Private Room "Ader" at BPS Hostel	0	Private Rooms at Bon Paul and Sharky's Hostel....	Private Rooms at Bon Paul and Sharky's Hostel....	746673	BonPaul	2011-

5 rows × 59 columns

In [8]:
```
# Changing the data types of columns to proper format
# 1. Change the host_since column to date format

listing_detail_df_new['host_since'] = listing_detail_df_new['host_since'].astype
```

In [9]:
```
#2.Change the zipcode column to int from float

listing_detail_df_new['zipcode'] = listing_detail_df_new['zipcode'].astype('int'
```

In [10]:
```
# 3. Change the price column to integer/float from object
# Remove special characters like $ and ,

listing_detail_df_new['price'] = listing_detail_df_new['price'].replace('[\$,]',
```

In [12]:
```
# 4. Convert the review scores columns from float to int

convert_datatype = {'review_scores_rating':int,
                    'review_scores_accuracy': int,
                    'review_scores_cleanliness':int,
                    'review_scores_checkin':int,
                    'review_scores_communication':int,
                    'review_scores_location':int,
                    'review_scores_value':int
                   }

listing_detail_df_new = listing_detail_df_new.astype(convert_datatype)
```

In [ ]:
```
listing_detail_df_new['price'] = listing_detail_df_new['price'].replace('[\$,]',
```

In [13]:
```
#Verify the data types are correct for all columns

listing_detail_df_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2387 entries, 0 to 2386
Data columns (total 59 columns):
 #   Column                          Non-Null Count  Dtype         
---  ------                          --------------  -----         
 0   id                              2387 non-null   int64         
 1   name                            2387 non-null   object        
 2   summary                         2387 non-null   object        
 3   space                           2387 non-null   object        
 4   description                     2387 non-null   object        
 5   host_id                         2387 non-null   int64         
 6   host_name                       2387 non-null   object        
 7   host_since                      2387 non-null   datetime64[ns]
 8   host_location                   2387 non-null   object        
 9   host_response_time              2387 non-null   object        
 10  host_response_rate              2387 non-null   object        
 11  host_acceptance_rate            2387 non-null   object        
 12  host_is_superhost               2387 non-null   object        
 13  host_listings_count             2387 non-null   int64         
 14  host_total_listings_count       2387 non-null   int64         
 15  street                          2387 non-null   object        
 16  neighbourhood_cleansed          2387 non-null   int64         
 17  city                            2387 non-null   object        
 18  state                           2387 non-null   object        
 19  zipcode                         2387 non-null   int32         
 20  market                          2387 non-null   object        
 21  smart_location                  2387 non-null   object        
 22  country_code                    2387 non-null   object        
 23  country                         2387 non-null   object        
 24  latitude                        2387 non-null   float64       
 25  longitude                       2387 non-null   float64       
 26  property_type                   2387 non-null   object        
 27  room_type                       2387 non-null   object        
 28  amenities                       2387 non-null   object        
 29  price                           2387 non-null   float64       
 30  minimum_nights                  2387 non-null   int64         
 31  maximum_nights                  2387 non-null   int64         
 32  minimum_minimum_nights          2387 non-null   int64         
 33  maximum_minimum_nights          2387 non-null   int64         
 34  minimum_maximum_nights          2387 non-null   int64         
 35  maximum_maximum_nights          2387 non-null   int64         
 36  minimum_nights_avg_ntm          2387 non-null   float64       
 37  maximum_nights_avg_ntm          2387 non-null   float64       
 38  calendar_updated                2387 non-null   object        
 39  has_availability                2387 non-null   object        
 40  availability_30                 2387 non-null   int64         
 41  availability_60                 2387 non-null   int64         
 42  availability_90                 2387 non-null   int64         
 43  availability_365                2387 non-null   int64         
 44  calendar_last_scraped           2387 non-null   object        
 45  number_of_reviews               2387 non-null   int64         
 46  number_of_reviews_ltm           2387 non-null   int64         
 47  first_review                    2387 non-null   object        
```

```
 48  last_review                     2387 non-null   object
 49  review_scores_rating            2387 non-null   int32
 50  review_scores_accuracy          2387 non-null   int32
 51  review_scores_cleanliness       2387 non-null   int32
 52  review_scores_checkin           2387 non-null   int32
 53  review_scores_communication     2387 non-null   int32
 54  review_scores_location          2387 non-null   int32
 55  review_scores_value             2387 non-null   int32
 56  cancellation_policy             2387 non-null   object
 57  calculated_host_listings_count  2387 non-null   int64
 58  reviews_per_month               2387 non-null   float64
dtypes: datetime64[ns](1), float64(6), int32(8), int64(18), object(26)
memory usage: 1.0+ MB
```

In [14]:
```
# Storing the above dataframe as a CSV file

listing_detail_df_new = listing_detail_df_new.to_csv('static/data/listing_detail_
```

In [16]:
```
# Establish connection to postgres database

engine = create_engine('postgres://aipqvzakwuyayg:b2ada3ef206b1daa65925a6a739523
connection = engine.connect()
```

In [17]:
```
engine.table_names()
```

Out[17]:
```
['host_details', 'listing_rating_count']
```

In [ ]:
```
listing_detail_grouped.to_sql(name='listing_rating_count', con=engine, if_exists=
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```
listing_detail = listing_detail_df[['zipcode','review_scores_rating']]
```

In [5]:
```
listing_detail ['review_score_normalized'] = listing_detail['review_scores_rating
```

```
C:\Users\conne\Anaconda3\envs\PythonData\lib\site-packages\ipykernel_launcher.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  """Entry point for launching an IPython kernel.
```

In [6]:
```
listing_detail.head()
```

Out[6]:

	zipcode	review_scores_rating	review_score_normalized
0	28804.0	96.0	4.8
1	28801.0	96.0	4.8
2	28801.0	90.0	4.5
3	28806.0	90.0	4.5
4	28806.0	90.0	4.5

In [7]:
```
bins = [0,1,2,3,4,5]

group_names = ["0-1 star","1-2 star","2-3 star","3-4 star","4-5 star"]
```

In [8]:
```
listing_detail['review_score_group'] = pd.cut(listing_detail['review_score_norma
```

```
C:\Users\conne\Anaconda3\envs\PythonData\lib\site-packages\ipykernel_launcher.p
y:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  """Entry point for launching an IPython kernel.
```

In [46]:
```
listing_detail.head()
```

Out[46]:

	zipcode	review_scores_rating	review_score_normalized	review_score_group
0	28804.0	96.0	4.8	4-5 star
1	28801.0	96.0	4.8	4-5 star
2	28801.0	90.0	4.5	4-5 star
3	28806.0	90.0	4.5	4-5 star
4	28806.0	90.0	4.5	4-5 star

In [9]:
```
listing_detail_grouped = listing_detail.groupby(['zipcode','review_score_group'])
```

In [10]:
```
listing_detail_grouped.reset_index(inplace=True)
```

In [11]:
```
listing_detail_grouped.head()
```

Out[11]:

	zipcode	review_score_group	review_scores_rating	review_score_normalized
0	28701.0	0-1 star	NaN	NaN
1	28701.0	1-2 star	NaN	NaN
2	28701.0	2-3 star	NaN	NaN
3	28701.0	3-4 star	NaN	NaN
4	28701.0	4-5 star	NaN	NaN

In [12]:
```
listing_detail_grouped = listing_detail_grouped.astype({"zipcode": int})
```

In [13]:
```
# listing_detail_grouped.fillna(0)
import numpy as np

listing_detail_grouped.replace(to_replace = np.nan, value =0,inplace=True)
```

In [14]:
```
listing_detail_grouped.columns
```

Out[14]:
```
Index(['zipcode', 'review_score_group', 'review_scores_rating',
       'review_score_normalized'],
      dtype='object')
```

In [15]:
```
listing_detail_grouped=listing_detail_grouped.astype({'review_scores_rating': in
```

In [16]:
```
listing_detail_grouped=listing_detail_grouped.astype({'review_score_normalized':
```

In [17]:
```
listing_detail_grouped=listing_detail_grouped.astype({'review_score_group': str
```

In [18]:
```
listing_detail_grouped.head()
```

Out[18]:

	zipcode	review_score_group	review_scores_rating	review_score_normalized
0	28701	0-1 star	0	0
1	28701	1-2 star	0	0
2	28701	2-3 star	0	0
3	28701	3-4 star	0	0
4	28701	4-5 star	0	0

In [19]:
```
listing_detail_grouped.dtypes
```

Out[19]:
```
zipcode                   int32
review_score_group       object
review_scores_rating      int32
review_score_normalized   int32
dtype: object
```

In [43]:
```
listing_rating_count = listing_detail_grouped.to_csv('static/data/listing_rating_
```

In [ ]:
```

```

In [22]:
```
from sqlalchemy import create_engine

engine = create_engine('postgres://aipqvzakwuyayg:b2ada3ef206b1daa65925a6a739523
connection = engine.connect()
```

In [23]:
```
# from sqlalchemy.ext.automap import automap_base
# from sqlalchemy.orm import Session
# from sqlalchemy import create_engine
# Base = automap_base()

engine.table_names()
```

Out[23]:
```
['listing_rating_count']
```

In [44]:
```
listing_detail_grouped.to_sql(name='listing_rating_count', con=engine, if_exists
```

```
---------------------------------------------------------------------------
UndefinedColumn                           Traceback (most recent call last)
~\Anaconda3\envs\PythonData\lib\site-packages\sqlalchemy\engine\base.py in _e
xecute_context(self, dialect, constructor, statement, parameters, *args)
   1227                     self.dialect.do_executemany(
-> 1228                         cursor, statement, parameters, context
   1229                     )

~\Anaconda3\envs\PythonData\lib\site-packages\sqlalchemy\dialects\postgresql
\psycopg2.py in do_executemany(self, cursor, statement, parameters, context)
    856         if self.executemany_mode is EXECUTEMANY_DEFAULT:
--> 857             cursor.executemany(statement, parameters)
    858             return

UndefinedColumn: column "index" of relation "listing_rating_count" does not e
xist
LINE 1: INSERT INTO listing_rating_count (index, zipcode, review_sco...
                                          ^
```

In [31]:
```
#For growth analysis

host_detail_df = listing_detail_df[['host_since']]
```

In [32]:
```
host_detail_df.head()
```

Out[32]:

	host_since
0	2010-07-13
1	2011-03-07
2	2010-12-16
3	2011-06-26
4	2011-06-26

In [33]:
```
host_detail_df.dtypes
```

Out[33]:
```
host_since    object
dtype: object
```

In [39]:
```
# host_detail_df=host_detail_df.astype({'host_since': date})

host_detail_df['host_since'] = host_detail_df['host_since'].astype('datetime64[ns
```

```
C:\Users\conne\Anaconda3\envs\PythonData\lib\site-packages\ipykernel_launcher.p
y:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  This is separate from the ipykernel package so we can avoid doing imports unt
il
```

In [40]:
```
host_detail_df.dtypes
```

Out[40]:
```
host_since    datetime64[ns]
dtype: object
```

In [45]:
```
host_details = host_detail_df.to_csv('static/data/host_details.csv',index=True)
```

In [41]:
```
host_detail_df.to_sql(name='host_details', con=engine, if_exists='append', index=
```

In [ ]:
```

```

In [16]:
```
# calendar_path = "templates/data/calendar.csv"

# calendar_df = pd.read_csv(calendar_path)
# calendar_df.head()
```

In [17]:
```
# listings_details_path = "templates/data/listings_details.csv"

# listings_details_df = pd.read_csv(listings_details_path)

# listings_details_df.rename(columns={'id':'listing_id'},inplace=True)

# listings_details_df.head(2)
```

In [18]:
```
# listings_details -

# listing_id,space,description,host_id,host_name,host_is_superhost, zipcode,lati
# security_deposit,cleaning_fee,extra_people,minimum_nights,maximum_nights,revie

# listings_d_need = listings_details_df.loc[:,['listing_id','space','description
# 'security_deposit','cleaning_fee','extra_people','minimum_nights','maximum_nigh

# listings_d_need.head()
```

In [19]:
```
# calendar_need =   calendar_df.loc[:,['listing_id','date','available']]

# calendar_need.head()
```

In [20]:
```
# calendar_listing = pd.concat([listings_d_need,calendar_need],axis=1,join='outer
```

In [21]:
```
# calendar_listing.head()
```

In [22]:
```
# calendar_listing.to_csv('templates/data/calendar_listing.csv',index=False)
```

In [23]:
```
# listings_path = "templates/data/listings.csv"

# listings_df = pd.read_csv(listings_path)

# listings_df.head(5)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [25]:
```
# listing_detail.head()
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [26]:
```
# list_d_path = "templates/data/listings_details.csv"

# listings_details_df = pd.read_csv(list_d_path)
# listings_details = listings_details_df[['zipcode','review_scores_rating']]
```

In [27]:
```
# listing_d = listings_details_df['review_scores_rating']
# listing_d.head()
```

In [28]:
```
# listings_merged = pd.concat([listings_df,listing_d],axis=1,join='outer')
```

In [29]:
```
# listings_merged.head()
```

In [30]:
```
# listings_merged ['review_score_normalized'] = listings_merged['review_scores_r
```

In [31]:
```
# listings_merged.head()
```

In [ ]:

In [ ]:

In [32]:
```
# bins = [0,1,2,3,4,5]

# group_names = ["0-1","1-2","2-3","3-4","4-5"]
```

In [33]:
```
# listings_merged['review_score_group'] = pd.cut(listings_merged['review_score_n
```

In [34]:
```
# listings_merged.head()
```

In [35]:
```
# listing_grouped = listings_merged.groupby(['neighbourhood','review_score_group
```

In [36]:
```
# listing_grouped
```

In [37]:
```
# listing_grouped.to_csv('templates/data/listing_rating_count.csv',index=True)
```

In [38]:
```
# listing_grouped = listings_merged.groupby(listings_merged['neighbourhood','rev
#                                                            'neighbourhood' : ['c
#                                                             'review_score_group'
#                                                          })
```

In [ ]:

In [ ]: