Airbnb_data_preprocessing-v2

April 25, 2021

```python
[269]: import pandas as pd
       import numpy as np
       import seaborn as sns
       from scipy import stats
       from sklearn import preprocessing
       import geohash as gh
       from sklearn.model_selection import train_test_split
       import category_encoders as ce
```

```python
[270]: df = pd.read_csv("Airbnb_NYC_2019.csv")
```

```python
[271]: df.head()
```

```
[271]:      id                                          name   host_id  \
       0  2539                 Clean & quiet apt home by the park     2787
       1  2595                             Skylit Midtown Castle     2845
       2  3647              THE VILLAGE OF HARLEM…NEW YORK !      4632
       3  3831                 Cozy Entire Floor of Brownstone     4869
       4  5022  Entire Apt: Spacious Studio/Loft by central park     7192

         host_name neighbourhood_group neighbourhood  latitude  longitude  \
       0       John             Brooklyn    Kensington  40.64749  -73.97237
       1   Jennifer            Manhattan       Midtown  40.75362  -73.98377
       2  Elisabeth            Manhattan        Harlem  40.80902  -73.94190
       3 LisaRoxanne            Brooklyn  Clinton Hill  40.68514  -73.95976
       4      Laura            Manhattan   East Harlem  40.79851  -73.94399

                 room_type  price  minimum_nights  number_of_reviews last_review  \
       0      Private room    149               1                  9  2018-10-19
       1  Entire home/apt    225               1                 45  2019-05-21
       2      Private room    150               3                  0         NaN
       3  Entire home/apt     89               1                270  2019-07-05
       4  Entire home/apt     80              10                  9  2018-11-19

          reviews_per_month  calculated_host_listings_count  availability_365
       0               0.21                               6               365
       1               0.38                               2               355
       2                NaN                               1               365
```

1

3	4.64	1	194
4	0.10	1	0

```
[272]: df.dtypes
```

```
[272]: id                                int64
       name                             object
       host_id                           int64
       host_name                        object
       neighbourhood_group              object
       neighbourhood                    object
       latitude                        float64
       longitude                       float64
       room_type                        object
       price                             int64
       minimum_nights                    int64
       number_of_reviews                 int64
       last_review                      object
       reviews_per_month               float64
       calculated_host_listings_count    int64
       availability_365                  int64
       dtype: object
```

```
[273]: df.shape
```

```
[273]: (48895, 16)
```

1   Dealing with missing values

```
[274]: # check missing values
       df.isna().sum()
```

```
[274]: id                                 0
       name                              16
       host_id                            0
       host_name                         21
       neighbourhood_group                0
       neighbourhood                      0
       latitude                           0
       longitude                          0
       room_type                          0
       price                              0
       minimum_nights                     0
       number_of_reviews                  0
       last_review                    10052
       reviews_per_month              10052
       calculated_host_listings_count     0
```

```
availability_365                    0
dtype: int64
```

We can keep the missing values for `name`, `host_name`, since we are not going to using these variables in the model anyways. Even if we were, it may be worth it to keep them in the model to decide what the output of a null host_name would be. As for `last_review` and `reviews_per_month`, we believe that `last_month` is a variable that we would never include in the model. For `reviews_per_month`, we can replace all the missing values to 0, because 0 should be the correct value if a review has never been made for that listing.

The empty values for name, host_name, and last reviews can be dropped, since they seem non-menaingful to impute. We can replace the empty values for reviews per month with 0 values, becuase this means there is no review per month.

```python
[275]: df['reviews_per_month'] = df['reviews_per_month'].fillna(0.00)
       df['name'] = df['name'].fillna('')
       df['host_name'] = df['host_name'].fillna('')
```

```python
[276]: #check missing values again
       df.isna().sum()
```

```
[276]: id                                  0
       name                                0
       host_id                             0
       host_name                           0
       neighbourhood_group                 0
       neighbourhood                       0
       latitude                            0
       longitude                           0
       room_type                           0
       price                               0
       minimum_nights                      0
       number_of_reviews                   0
       last_review                     10052
       reviews_per_month                   0
       calculated_host_listings_count      0
       availability_365                    0
       dtype: int64
```

```python
[277]: df.describe()
```

```
[277]:                 id        host_id      latitude     longitude         price  \
       count  4.889500e+04  4.889500e+04  48895.000000  48895.000000  48895.000000
       mean   1.901714e+07  6.762001e+07     40.728949    -73.952170    152.720687
       std    1.098311e+07  7.861097e+07      0.054530      0.046157    240.154170
       min    2.539000e+03  2.438000e+03     40.499790    -74.244420      0.000000
       25%    9.471945e+06  7.822033e+06     40.690100    -73.983070     69.000000
       50%    1.967728e+07  3.079382e+07     40.723070    -73.955680    106.000000
```

```
75%    2.915218e+07  1.074344e+08    40.763115    -73.936275    175.000000
max    3.648724e+07  2.743213e+08    40.913060    -73.712990  10000.000000

       minimum_nights  number_of_reviews  reviews_per_month  \
count     48895.000000       48895.000000       48895.000000
mean          7.029962          23.274466           1.090910
std          20.510550          44.550582           1.597283
min           1.000000           0.000000           0.000000
25%           1.000000           1.000000           0.040000
50%           3.000000           5.000000           0.370000
75%           5.000000          24.000000           1.580000
max        1250.000000         629.000000          58.500000

       calculated_host_listings_count  availability_365
count                    48895.000000      48895.000000
mean                         7.143982        112.781327
std                         32.952519        131.622289
min                          1.000000          0.000000
25%                          1.000000          0.000000
50%                          1.000000         45.000000
75%                          2.000000        227.000000
max                        327.000000        365.000000
```

2 Omit extreme outliers and invalid values

There are also be some erroneous values in the dataset. For example, there are instances where the price is 10,000 per day despite being a single private room.

For `price`, we omit from the dataset if the price is above 3,000 per day or costs 0 per day. For `minimum_nights`, we omit if the number of minimum nights is above 60 days per month. For `reviews_per_month`, we omit if the number is above 15 per month, as it is very unlikely a listing could get 15 reviews a month, which is a review every 2 days.

```
[278]: df_omit = df[(df['price'] > 0) & (df['price'] <= 3000) & (df['minimum_nights']␣
       ↪<= 60) & (df['reviews_per_month'] <= 15)]
```

2.1 What are some components that need to take into considerations for house price?

geography (`latitude`, `longitude`), `minimum_nights`, `number_of_reviews`, `reviews_per_month`, `calculated_host_listings_count`, `availability_365`. Thus we can exclude `id`, `host_id`, `last_review` from our considerations for training data. We also exclude `neigbourhood_group` from our analysis as we believe this too closely overlaps with coordinate data.

```
[279]: #generate training data
       # drop unrelated information
       # neighborhood has the same information as latitude and longitude, thus␣
       ↪enighborhood can be dropped
```

```
df_relevant = df_omit.drop(['id','name', 'host_name',␣
 ↪'host_id','neighbourhood','last_review'], axis=1)
```

[280]: `df_relevant.head()`

[280]:
```
   neighbourhood_group  latitude  longitude        room_type  price  \
0              Brooklyn  40.64749  -73.97237     Private room    149
1             Manhattan  40.75362  -73.98377  Entire home/apt    225
2             Manhattan  40.80902  -73.94190     Private room    150
3              Brooklyn  40.68514  -73.95976  Entire home/apt     89
4             Manhattan  40.79851  -73.94399  Entire home/apt     80

   minimum_nights  number_of_reviews  reviews_per_month  \
0               1                  9               0.21
1               1                 45               0.38
2               3                  0               0.00
3               1                270               4.64
4              10                  9               0.10

   calculated_host_listings_count  availability_365
0                               6               365
1                               2               355
2                               1               365
3                               1               194
4                               1                 0
```

[281]: *#check number of unique values in each columne to decide what processing␣*
       ↪technique to use
```
df_relevant.nunique()
```

[281]:
```
neighbourhood_group                 5
latitude                        18995
longitude                       14674
room_type                           3
price                             639
minimum_nights                     50
number_of_reviews                 391
reviews_per_month                 927
calculated_host_listings_count     47
availability_365                  366
dtype: int64
```

3 One hot encoding for categorical variables

```
[282]: print(df_relevant['room_type'].unique())
       print(df_relevant['neighbourhood_group'].unique())
```

```
['Private room' 'Entire home/apt' 'Shared room']
['Brooklyn' 'Manhattan' 'Queens' 'Staten Island' 'Bronx']
```

Based on the number of unique values and data type for each column. We can apply the following encoding method for text preprocessing:

1. one hot encoding for neighbor group
2. create grouping for latitude and longitude first? then encode? 3.label encode for room type since size matters
3. conduct normalization/ standardization for all continuous data

```
[283]: #exclude labe
       df_relevant.drop(['price'], axis = 1, inplace= True)
```

```
[284]: df_relevant.head()
```

```
[284]:   neighbourhood_group  latitude  longitude        room_type  minimum_nights  \
       0             Brooklyn  40.64749  -73.97237      Private room               1
       1            Manhattan  40.75362  -73.98377  Entire home/apt               1
       2            Manhattan  40.80902  -73.94190      Private room               3
       3             Brooklyn  40.68514  -73.95976  Entire home/apt               1
       4            Manhattan  40.79851  -73.94399  Entire home/apt              10

          number_of_reviews  reviews_per_month  calculated_host_listings_count  \
       0                  9               0.21                               6
       1                 45               0.38                               2
       2                  0               0.00                               1
       3                270               4.64                               1
       4                  9               0.10                               1

          availability_365
       0               365
       1               355
       2               365
       3               194
       4                 0
```

```
[285]: df_relevant_encode = pd.get_dummies(df_relevant, prefix =␣
       ↪['neighbourhood_group', 'room_type'], columns = ['neighbourhood_group',␣
       ↪'room_type'])
       df_relevant_label = pd.get_dummies(df_relevant, prefix =␣
       ↪['neighbourhood_group'], columns = ['neighbourhood_group'])
       le = preprocessing.LabelEncoder()
       le.fit(['Shared room','Private room','Entire home/apt'])
```

```
df_relevant_label['room_type'] = le.transform(df_relevant_label['room_type'])
```

4 Geohash for latitude and longitude

```
[286]: # create geohash code for geographical data
       df_relevant_encode['geohash']=df_relevant.apply(lambda x: gh.
        ↪encode(x['latitude'], x['longitude'], precision=7), axis=1)
```

```
[287]: df_relevant_encode.head()
```

```
[287]:    latitude  longitude  minimum_nights  number_of_reviews  reviews_per_month  \
       0  40.64749  -73.97237               1                  9               0.21
       1  40.75362  -73.98377               1                 45               0.38
       2  40.80902  -73.94190               3                  0               0.00
       3  40.68514  -73.95976               1                270               4.64
       4  40.79851  -73.94399              10                  9               0.10

          calculated_host_listings_count  availability_365  \
       0                               6               365
       1                               2               355
       2                               1               365
       3                               1               194
       4                               1                 0

          neighbourhood_group_Bronx  neighbourhood_group_Brooklyn  \
       0                          0                             1
       1                          0                             0
       2                          0                             0
       3                          0                             1
       4                          0                             0

          neighbourhood_group_Manhattan  neighbourhood_group_Queens  \
       0                              0                           0
       1                              1                           0
       2                              1                           0
       3                              0                           0
       4                              1                           0

          neighbourhood_group_Staten Island  room_type_Entire home/apt  \
       0                                  0                          0
       1                                  0                          1
       2                                  0                          0
       3                                  0                          1
       4                                  0                          1

          room_type_Private room  room_type_Shared room  geohash
```

```
0                    1                    0   dr5rhxw
1                    0                    0   dr5ru6y
2                    1                    0   dr72jmj
3                    0                    0   dr5rmn8
4                    0                    0   dr72j75
```

[288]:
```python
#drop latltitude longitude
df_relevant.drop(['latitude', 'longitude'], axis = 1, inplace= True)
```

[289]:
```python
df_relevant.head()
```

[289]:
```
  neighbourhood_group        room_type  minimum_nights  number_of_reviews  \
0            Brooklyn     Private room               1                  9
1           Manhattan  Entire home/apt               1                 45
2           Manhattan     Private room               3                  0
3            Brooklyn  Entire home/apt               1                270
4           Manhattan  Entire home/apt              10                  9

   reviews_per_month  calculated_host_listings_count  availability_365
0               0.21                               6               365
1               0.38                               2               355
2               0.00                               1               365
3               4.64                               1               194
4               0.10                               1                 0
```

[290]:
```python
df_relevant_encode.geohash.nunique()
# there are 10442 unique geographical location, should apply target encoding␣
↪later
```

[290]: 10442

[294]:
```python
X_col_names = df_relevant_encode.columns
X = df_relevant_encode.values.tolist()
y = df_omit['price'].tolist()
```

5   Split Train and Test Data (2/3, 1/3 split)

Split train data and test data for this one, with 67% in the training set and 33% in the testing set.

[295]:
```python
#train test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,␣
↪random_state=42)
print(len(X_train))
print(len(X_test))
print(len(y_train))
print(len(y_test))
```

```
32495
16006
32495
16006
```

[296]:
```python
#create train and test dataframe for target encoding later
df_train = pd.DataFrame(X_train)
df_test = pd.DataFrame(X_test)
df_train.columns = X_col_names
df_test.columns = X_col_names
```

6 Transform Continuous Variables

Use `TargetEncoder` to encode the `geohash`. Also, transform the y-variable and x-variables if necessary into either normalized/standardized form.

Normalization is good to use when you know that the distribution of your data does not follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like K-Nearest Neighbors and Neural Networks.

Standardization, on the other hand, can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.

[297]:
```python
# target encode on geolocations, since the amount of unique values are large
# if we look at price as a target, each row with the unique value of␣
 ↪geolocation would be replaced with the average price for the house
encoder = ce.TargetEncoder(cols=['geohash'], smoothing=0, return_df=True)

df_train['coded_geo'] = encoder.fit_transform(df_train['geohash'], y_train)
df_test['coded_geo'] = encoder.transform(df_train['geohash'])
```

```
C:\Users\williamshih\anaconda3\lib\site-packages\category_encoders\utils.py:21:
FutureWarning: is_categorical is deprecated and will be removed in a future
version.  Use is_categorical_dtype instead
  elif pd.api.types.is_categorical(cols):
```

[298]:
```python
df_train.drop('geohash', axis=1, inplace= True)
df_test.drop('geohash', axis=1, inplace= True)
```

It turns out the y-variable could benefit from a log transformation, depending on what model we are using as the distribution of prices is close to a lognormal distribution.

[299]:
```python
one, two, three = stats.boxcox(y_train, alpha = 0.95)
print(three)
y_train = np.log(y_train)
y_test = np.log(y_test)
```

```
(-0.21245489163451545, -0.2116241203325513)
```

```python
[300]: # concatenate train and test dataframes again for normalization or␣
       ↪stanadardization
       df_train['price'] = y_train
       df_test['price'] = y_test
       df_whole = pd.concat([df_train, df_test])
```

```python
[301]: # apply standarization or normalization on continuous values based on the data␣
       ↪distribution
       to_scale = ['minimum_nights', 'number_of_reviews', 'reviews_per_month',
               'calculated_host_listings_count','coded_geo', 'availability_365','price']
       scaled_train = df_train.copy()
       scaled_test = df_test.copy()
       scaled_features = scaled_train[to_scale]
       scaler = preprocessing.StandardScaler().fit(scaled_features)
       scaled_train[to_scale] = scaler.transform(scaled_features)
       scaled_test[to_scale] = scaler.transform(scaled_test[to_scale])
```

```python
[312]: # This is extra code in case room_type uses the label encode instead of one-hot␣
       ↪encoding
       scaler2 = preprocessing.StandardScaler().fit(df_relevant_label['room_type'].
       ↪values.reshape(-1,1))
       df_relevant_label['room_type'] = scaler2.
       ↪transform(df_relevant_label['room_type'].values.reshape(-1,1))
```

```
0          0.907403
1         -0.926574
2          0.907403
3         -0.926574
4         -0.926574
             …
48890      0.907403
48891      0.907403
48892     -0.926574
48893      2.741380
48894      0.907403
Name: room_type, Length: 48501, dtype: float64
```

```python
[302]: print(scaler.mean_, scaler.var_)
       print(scaler2.mean_, scaler2.var_)
```

```
[  5.91137098  23.40089244   1.08921157   7.18729035 151.25913372
  112.18218187   4.72728818]
[7.92522157e+01 1.98228751e+03 2.39061991e+00 1.09447552e+03
 6.71232886e+03 1.72419016e+04 4.70491786e-01]
```

```python
[303]: scaled_train.describe()
```

```
[303]:           latitude      longitude    minimum_nights    number_of_reviews  \
       count   32495.000000  32495.000000      3.249500e+04         3.249500e+04
       mean       40.728897    -73.952048      4.564846e-16         7.190227e-17
       std         0.054579      0.046227      1.000015e+00         1.000015e+00
       min        40.499790    -74.244420     -5.516924e-01        -5.255924e-01
       25%        40.689930    -73.982910     -5.516924e-01        -5.031321e-01
       50%        40.722920    -73.955650     -4.393628e-01        -4.132906e-01
       75%        40.763115    -73.936275     -1.023740e-01         1.345617e-02
       max        40.913060    -73.712990      6.075755e+00         1.310784e+01


              reviews_per_month   calculated_host_listings_count   availability_365  \
       count       3.249500e+04                     3.249500e+04       3.249500e+04
       mean        4.014877e-16                     4.159160e-16      -1.299673e-16
       std         1.000015e+00                     1.000015e+00       1.000015e+00
       min        -7.044610e-01                    -1.870241e-01      -8.543411e-01
       25%        -6.785905e-01                    -1.870241e-01      -8.543411e-01
       50%        -4.586913e-01                    -1.870241e-01      -5.192522e-01
       75%         3.238911e-01                    -1.567969e-01       8.515662e-01
       max         8.583046e+00                     9.667022e+00       1.925374e+00


              neighbourhood_group_Bronx   neighbourhood_group_Brooklyn  \
       count                32495.000000                   32495.000000
       mean                     0.022465                       0.412094
       std                      0.148192                       0.492219
       min                      0.000000                       0.000000
       25%                      0.000000                       0.000000
       50%                      0.000000                       0.000000
       75%                      0.000000                       1.000000
       max                      1.000000                       1.000000


              neighbourhood_group_Manhattan   neighbourhood_group_Queens  \
       count                   32495.000000                 32495.000000
       mean                        0.441268                     0.116664
       std                         0.496546                     0.321025
       min                         0.000000                     0.000000
       25%                         0.000000                     0.000000
       50%                         0.000000                     0.000000
       75%                         1.000000                     0.000000
       max                         1.000000                     1.000000


              neighbourhood_group_Staten Island   room_type_Entire home/apt  \
       count                        32495.000000                32495.000000
       mean                             0.007509                    0.520819
       std                              0.086329                    0.499574
       min                              0.000000                    0.000000
       25%                              0.000000                    0.000000
       50%                              0.000000                    1.000000
```

```
75%                              0.000000                   1.000000
max                              1.000000                   1.000000


        room_type_Private room  room_type_Shared room       coded_geo  \
count              32495.000000            32495.000000    3.249500e+04
mean                   0.455485                0.023696    1.251949e-16
std                    0.498022                0.152102    1.000015e+00
min                    0.000000                0.000000   -1.602112e+00
25%                    0.000000                0.000000   -6.500660e-01
50%                    0.000000                0.000000   -5.122293e-02
75%                    1.000000                0.000000    3.499311e-01
max                    1.000000                1.000000    1.886688e+01


                price
count    3.249500e+04
mean     1.184045e-15
std      1.000015e+00
min     -3.534944e+00
25%     -7.190033e-01
50%     -6.583376e-02
75%      6.461296e-01
max      4.780528e+00
```

[304]: `scaled_test.describe()`

[304]:
```
            latitude     longitude   minimum_nights   number_of_reviews  \
count   16006.000000  16006.000000     16006.000000        16006.000000
mean       40.728899    -73.952187        -0.008719           -0.004762
std         0.054471      0.046053         0.995663            0.995522
min        40.508730    -74.239140        -0.551692           -0.525592
25%        40.690252    -73.983130        -0.551692           -0.503132
50%        40.723115    -73.955635        -0.439363           -0.413291
75%        40.763130    -73.936050        -0.102374            0.013456
max        40.908040    -73.718290         6.075755           13.601973


        reviews_per_month  calculated_host_listings_count  availability_365  \
count        16006.000000                    16006.000000      16006.000000
mean             0.006815                       -0.004808         -0.001214
std              1.025670                        0.988652          1.000846
min             -0.704461                       -0.187024         -0.854341
25%             -0.678591                       -0.187024         -0.854341
50%             -0.465159                       -0.187024         -0.519252
75%              0.343294                       -0.156797          0.874413
max              8.751205                        9.667022          1.925374


        neighbourhood_group_Bronx  neighbourhood_group_Brooklyn  \
count                16006.000000                  16006.000000
```

```
mean                                    0.021742                          0.411346
std                                     0.145844                          0.492093
min                                     0.000000                          0.000000
25%                                     0.000000                          0.000000
50%                                     0.000000                          0.000000
75%                                     0.000000                          1.000000
max                                     1.000000                          1.000000

       neighbourhood_group_Manhattan  neighbourhood_group_Queens  \
count                    16006.000000                16006.000000
mean                         0.443584                    0.115394
std                          0.496823                    0.319507
min                          0.000000                    0.000000
25%                          0.000000                    0.000000
50%                          0.000000                    0.000000
75%                          1.000000                    0.000000
max                          1.000000                    1.000000

       neighbourhood_group_Staten Island  room_type_Entire home/apt  \
count                        16006.000000                16006.00000
mean                             0.007935                    0.51362
std                              0.088725                    0.49983
min                              0.000000                    0.00000
25%                              0.000000                    0.00000
50%                              0.000000                    1.00000
75%                              0.000000                    1.00000
max                              1.000000                    1.00000

       room_type_Private room  room_type_Shared room     coded_geo  \
count            16006.000000           16006.000000  16006.000000
mean                 0.462764               0.023616     -0.003996
std                  0.498627               0.151855      1.021683
min                  0.000000               0.000000     -1.565495
25%                  0.000000               0.000000     -0.656169
50%                  0.000000               0.000000     -0.064192
75%                  1.000000               0.000000      0.346734
max                  1.000000               1.000000     18.866880

              price
count  16006.000000
mean      -0.018811
std        1.006291
min       -3.534944
25%       -0.740287
50%       -0.106904
75%        0.637823
max        4.780528
```

```
[305]:  # after nromalization and standardization, split the data into train and test
        →with the same proportion as before
```

```
[208]:  # correlation plot to decide variables
        scaled_train.corr()
```

```
[208]:                                       minimum_nights  number_of_reviews  \
        minimum_nights                             1.000000          -0.151380
        number_of_reviews                         -0.151380           1.000000
        reviews_per_month                         -0.227260           0.598994
        calculated_host_listings_count             0.304562          -0.073046
        availability_365                           0.238644           0.173356
        neighbourhood_group_Bronx                 -0.044398           0.009050
        neighbourhood_group_Brooklyn              -0.071559           0.018435
        neighbourhood_group_Manhattan              0.121706          -0.043514
        neighbourhood_group_Queens                -0.053303           0.032030
        neighbourhood_group_Staten Island         -0.017594           0.010530
        room_type_Entire home/apt                  0.134518          -0.008860
        room_type_Private room                    -0.130357           0.015882
        room_type_Shared room                     -0.014994          -0.022902
        coded_geo                                  0.093063          -0.048141
        price                                      0.047964          -0.056905

                                          reviews_per_month  \
        minimum_nights                            -0.227260
        number_of_reviews                          0.598994
        reviews_per_month                          1.000000
        calculated_host_listings_count            -0.046984
        availability_365                           0.167878
        neighbourhood_group_Bronx                  0.039902
        neighbourhood_group_Brooklyn              -0.019298
        neighbourhood_group_Manhattan             -0.064425
        neighbourhood_group_Queens                 0.102513
        neighbourhood_group_Staten Island          0.030888
        room_type_Entire home/apt                 -0.025827
        room_type_Private room                     0.025508
        room_type_Shared room                      0.001310
        coded_geo                                 -0.062643
        price                                     -0.059836

                                          calculated_host_listings_count  \
        minimum_nights                                          0.304562
        number_of_reviews                                      -0.073046
        reviews_per_month                                      -0.046984
        calculated_host_listings_count                          1.000000
        availability_365                                        0.229548
        neighbourhood_group_Bronx                              -0.022897
```

```
neighbourhood_group_Brooklyn                          -0.123258
neighbourhood_group_Manhattan                          0.153705
neighbourhood_group_Queens                            -0.034818
neighbourhood_group_Staten Island                     -0.012518
room_type_Entire home/apt                              0.111129
room_type_Private room                                -0.108264
room_type_Shared room                                 -0.010514
coded_geo                                              0.157651
price                                                  0.096774


                                   availability_365  \
minimum_nights                             0.238644
number_of_reviews                          0.173356
reviews_per_month                          0.167878
calculated_host_listings_count             0.229548
availability_365                           1.000000
neighbourhood_group_Bronx                  0.065808
neighbourhood_group_Brooklyn              -0.082752
neighbourhood_group_Manhattan             -0.005481
neighbourhood_group_Queens                 0.089800
neighbourhood_group_Staten Island          0.056456
room_type_Entire home/apt                 -0.010676
room_type_Private room                    -0.007608
room_type_Shared room                      0.059974
coded_geo                                  0.052003
price                                      0.106367


                                   neighbourhood_group_Bronx  \
minimum_nights                                     -0.044398
number_of_reviews                                   0.009050
reviews_per_month                                   0.039902
calculated_host_listings_count                     -0.022897
availability_365                                    0.065808
neighbourhood_group_Bronx                           1.000000
neighbourhood_group_Brooklyn                       -0.126920
neighbourhood_group_Manhattan                      -0.134721
neighbourhood_group_Queens                         -0.055093
neighbourhood_group_Staten Island                  -0.013186
room_type_Entire home/apt                          -0.051628
room_type_Private room                              0.040654
room_type_Shared room                              0.036457
coded_geo                                          -0.039419
price                                              -0.055643


                                   neighbourhood_group_Brooklyn  \
minimum_nights                                        -0.071559
number_of_reviews                                      0.018435
```

```
reviews_per_month                           -0.019298
calculated_host_listings_count              -0.123258
availability_365                            -0.082752
neighbourhood_group_Bronx                   -0.126920
neighbourhood_group_Brooklyn                 1.000000
neighbourhood_group_Manhattan               -0.744036
neighbourhood_group_Queens                  -0.304264
neighbourhood_group_Staten Island           -0.072823
room_type_Entire home/apt                   -0.074125
room_type_Private room                       0.078789
room_type_Shared room                       -0.014515
coded_geo                                   -0.278999
price                                       -0.147146


                                   neighbourhood_group_Manhattan  \
minimum_nights                                        0.121706
number_of_reviews                                    -0.043514
reviews_per_month                                    -0.064425
calculated_host_listings_count                        0.153705
availability_365                                     -0.005481
neighbourhood_group_Bronx                            -0.134721
neighbourhood_group_Brooklyn                         -0.744036
neighbourhood_group_Manhattan                         1.000000
neighbourhood_group_Queens                           -0.322965
neighbourhood_group_Staten Island                    -0.077299
room_type_Entire home/apt                             0.161154
room_type_Private room                               -0.158946
room_type_Shared room                                -0.008873
coded_geo                                             0.398943
price                                                 0.244697


                                   neighbourhood_group_Queens  \
minimum_nights                                      -0.053303
number_of_reviews                                    0.032030
reviews_per_month                                    0.102513
calculated_host_listings_count                      -0.034818
availability_365                                     0.089800
neighbourhood_group_Bronx                           -0.055093
neighbourhood_group_Brooklyn                        -0.304264
neighbourhood_group_Manhattan                       -0.322965
neighbourhood_group_Queens                           1.000000
neighbourhood_group_Staten Island                   -0.031610
room_type_Entire home/apt                           -0.109460
room_type_Private room                               0.104187
room_type_Shared room                                0.018384
coded_geo                                           -0.163495
price                                               -0.119664
```

```
                                neighbourhood_group_Staten Island  \
minimum_nights                                          -0.017594
number_of_reviews                                        0.010530
reviews_per_month                                        0.030888
calculated_host_listings_count                          -0.012518
availability_365                                         0.056456
neighbourhood_group_Bronx                               -0.013186
neighbourhood_group_Brooklyn                            -0.072823
neighbourhood_group_Manhattan                           -0.077299
neighbourhood_group_Queens                              -0.031610
neighbourhood_group_Staten Island                        1.000000
room_type_Entire home/apt                               -0.008620
room_type_Private room                                   0.007775
room_type_Shared room                                    0.002855
coded_geo                                               -0.028234
price                                                   -0.027971

                                room_type_Entire home/apt  \
minimum_nights                                   0.134518
number_of_reviews                               -0.008860
reviews_per_month                               -0.025827
calculated_host_listings_count                   0.111129
availability_365                                -0.010676
neighbourhood_group_Bronx                       -0.051628
neighbourhood_group_Brooklyn                    -0.074125
neighbourhood_group_Manhattan                    0.161154
neighbourhood_group_Queens                      -0.109460
neighbourhood_group_Staten Island               -0.008620
room_type_Entire home/apt                        1.000000
room_type_Private room                          -0.953511
room_type_Shared room                           -0.162419
coded_geo                                        0.280548
price                                            0.381120

                                room_type_Private room  \
minimum_nights                                -0.130357
number_of_reviews                              0.015882
reviews_per_month                              0.025508
calculated_host_listings_count                -0.108264
availability_365                              -0.007608
neighbourhood_group_Bronx                      0.040654
neighbourhood_group_Brooklyn                   0.078789
neighbourhood_group_Manhattan                 -0.158946
neighbourhood_group_Queens                     0.104187
neighbourhood_group_Staten Island              0.007775
room_type_Entire home/apt                     -0.953511
```

```
room_type_Private room                               1.000000
room_type_Shared room                               -0.142488
coded_geo                                           -0.264835
price                                               -0.358139


                                   room_type_Shared room  coded_geo      price
minimum_nights                              -0.014994   0.093063   0.047964
number_of_reviews                           -0.022902  -0.048141  -0.056905
reviews_per_month                            0.001310  -0.062643  -0.059836
calculated_host_listings_count              -0.010514   0.157651   0.096774
availability_365                             0.059974   0.052003   0.106367
neighbourhood_group_Bronx                    0.036457  -0.039419  -0.055643
neighbourhood_group_Brooklyn                -0.014515  -0.278999  -0.147146
neighbourhood_group_Manhattan               -0.008873   0.398943   0.244697
neighbourhood_group_Queens                   0.018384  -0.163495  -0.119664
neighbourhood_group_Staten Island            0.002855  -0.028234  -0.027971
room_type_Entire home/apt                   -0.162419   0.280548   0.381120
room_type_Private room                      -0.142488  -0.264835  -0.358139
room_type_Shared room                        1.000000  -0.054311  -0.079134
coded_geo                                   -0.054311   1.000000   0.541421
price                                       -0.079134   0.541421   1.000000
```

[ ]: