Lab 1 Exploratory Analysis - Problems

February 7, 2022

1 Exploratory Analysis

1.1 Problems:

Load the NYC AirBnB Truncated Dataset. This dataset is a mirror of the full NYC AirBnB dataset found at Kaggle, but only contains the first 10,000 entries.

https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data

For the numerical features,

1) Display histograms for the numerical features.
2) Construct the scatter plots of price with each of the numerical features.
3) Display the correlation histogram.
4) Using numerical features to predict the renting price.
5) Write down the predict function from (4)
6) Calculate the RSS cost.

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

data = pd.read_csv("https://raw.githubusercontent.com/tipthederiver/
↪Math-7243-2020/master/Datasets/NYCAirBnB/train.csv")
data
```

[1]:
```
                                                 name   host_id    host_name  \
0                    Clean & quiet apt home by the park      2787         John
1                                 Skylit Midtown Castle      2845     Jennifer
2                   THE VILLAGE OF HARLEM…NEW YORK !      4632    Elisabeth
3                       Cozy Entire Floor of Brownstone      4869  LisaRoxanne
4      Entire Apt: Spacious Studio/Loft by central park      7192        Laura
…                                                   …         …            …
9994                 Cozy apt in heart of the e village  40076332       Steven
9995          Perfect Location - Meticulously Kept Flat  12620454         Will
9996                 Garden Apt in Historic Brownstone!   2060383         Lisa
```

```
9997              East Village Private Room & Terrace  39956905         Can
9998                Cosy apartment in Carroll Gardens  33064750       Suzan

     neighbourhood_group    neighbourhood  latitude  longitude  \
0               Brooklyn       Kensington  40.64749  -73.97237
1              Manhattan          Midtown  40.75362  -73.98377
2              Manhattan           Harlem  40.80902  -73.94190
3               Brooklyn     Clinton Hill  40.68514  -73.95976
4              Manhattan      East Harlem  40.79851  -73.94399
…                    …              …       …        …
9994           Manhattan     East Village  40.72644  -73.98403
9995            Brooklyn         Bushwick  40.70442  -73.92484
9996            Brooklyn      Cobble Hill  40.68732  -73.99245
9997           Manhattan     East Village  40.72811  -73.98453
9998            Brooklyn  Carroll Gardens  40.68282  -73.99774

             room_type  price  minimum_nights  number_of_reviews last_review  \
0         Private room    149               1                  9  10/19/2018
1      Entire home/apt    225               1                 45   5/21/2019
2         Private room    150               3                  0         NaN
3      Entire home/apt     89               1                270    7/5/2019
4      Entire home/apt     80              10                  9  11/19/2018
…                  …    …             …                …         …
9994   Entire home/apt    175               5                  0         NaN
9995   Entire home/apt    220               5                 27    1/1/2017
9996   Entire home/apt    147               3                 23   6/16/2019
9997      Private room     95               2                  1   8/29/2015
9998   Entire home/apt    160               5                  2    8/8/2017

      reviews_per_month  calculated_host_listings_count  availability_365
0                  0.21                               6               365
1                  0.38                               2               355
2                   NaN                               1               365
3                  4.64                               1               194
4                  0.10                               1                 0
…                   …                              …               …
9994                NaN                               1                 0
9995               0.57                               1                 0
9996               0.51                               1                 2
9997               0.02                               2                 0
9998               0.06                               1                 0

[9999 rows x 15 columns]
```

[2]:
```
data.size
```

[2]:
```
149985
```

[3]:
```
data.dtypes
```

[3]:
```
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
```

[4]:
```
data.isnull().sum()
```

[4]:
```
name                                 8
host_id                              0
host_name                           10
neighbourhood_group                  0
neighbourhood                        0
latitude                             0
longitude                            0
room_type                            0
price                                0
minimum_nights                       0
number_of_reviews                    0
last_review                       1322
reviews_per_month                 1322
calculated_host_listings_count       0
availability_365                     0
dtype: int64
```

[5]:
```
data = data.dropna()
```

[6]:
```
data.isnull().sum()
```

[6]:
```
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                       0
reviews_per_month                 0
calculated_host_listings_count    0
availability_365                  0
dtype: int64
```

[7]:
```
cleaned_data = data.drop(columns=['name',
                                  'host_id',
                                  'host_name',
                                  'neighbourhood_group',
                                  'neighbourhood',
                                  'room_type',
                                  'last_review',
                                  'latitude',
                                  'longitude'])
```

[8]:
```
cleaned_data.dtypes
```

[8]:
```
price                               int64
minimum_nights                      int64
number_of_reviews                   int64
reviews_per_month                 float64
calculated_host_listings_count      int64
availability_365                    int64
dtype: object
```

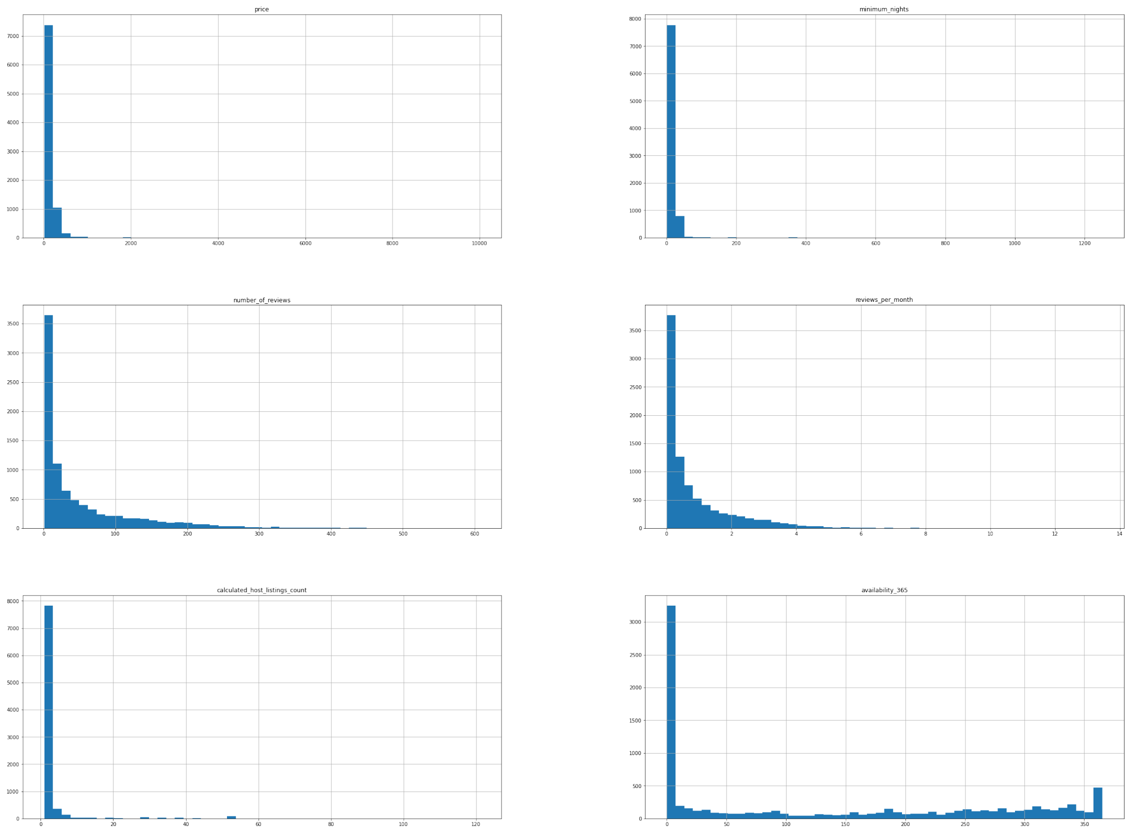
[9]:
```
cleaned_data.size
```

[9]:
```
51996
```

[10]:
```
X = cleaned_data.drop(columns='price').values
Y = cleaned_data['price'].values
ones = np.ones(X.shape[0]).reshape((-1, 1))
X = np.concatenate((ones, X), axis=1)
```

2 Histogram

[11]:
```python
cleaned_data.hist(bins=50, figsize=(40, 30))
plt.show()
```



[12]:
```python
names = list(cleaned_data)
X[:,1]
```

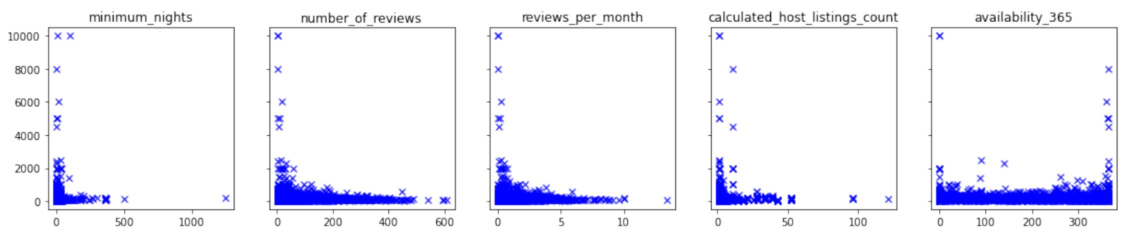
[12]:
```
array([1., 1., 1., …, 3., 2., 5.])
```

[13]:
```python
import collections

collections.Counter(X[:, 0])
```

[13]:
```
Counter({1.0: 8666})
```

3 ScatterPlot

[14]:
```
f, axes = plt.subplots(1, 5, sharey = True)
f.set_size_inches(15, 3)
f.tight_layout()

axes = axes.reshape(5)

for i in range(1, 6):
    axes[i - 1].plot(X[:, i], Y, 'x', color='Blue')
    axes[i - 1].set_title(names[i], fontsize=12)
```



4 Correlation

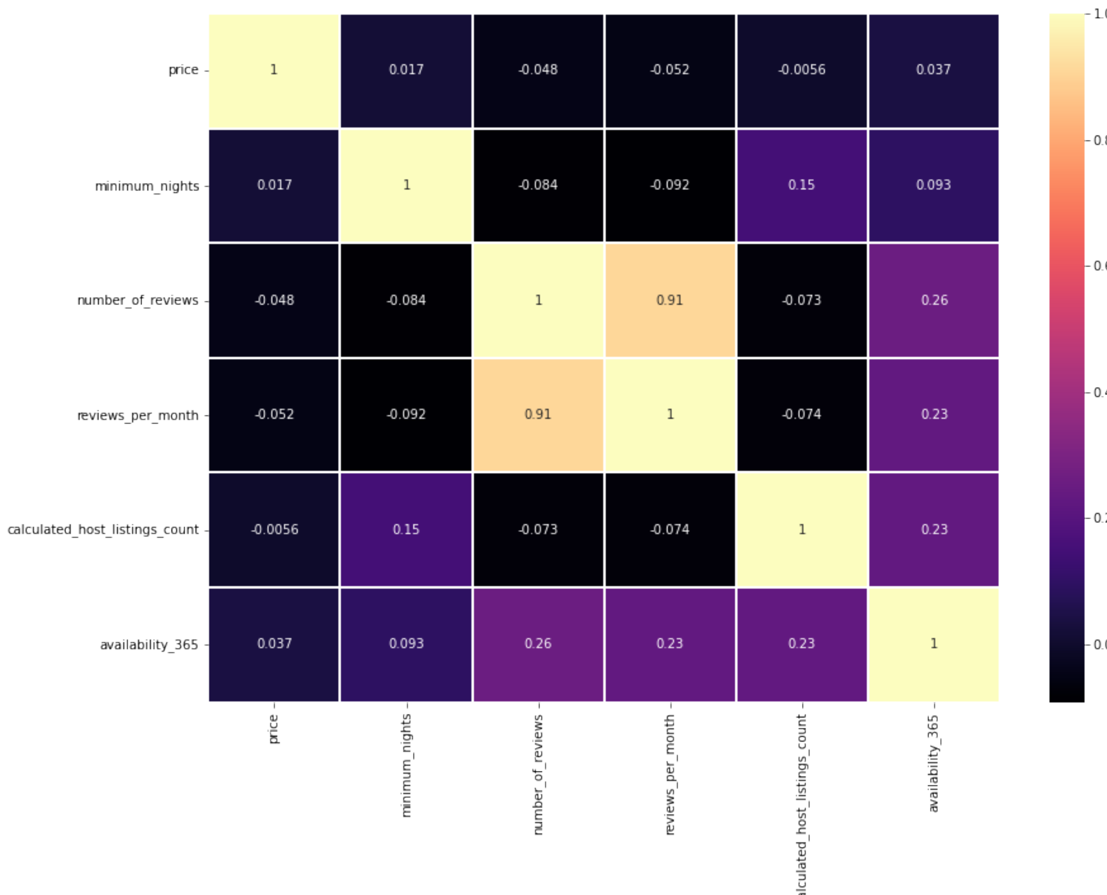
[15]:
```
correlation_matrix = cleaned_data.corr()
correlation_matrix
```

[15]:
```
                                  price  minimum_nights  number_of_reviews  \
price                          1.000000        0.017411          -0.048167
minimum_nights                 0.017411        1.000000          -0.084284
number_of_reviews             -0.048167       -0.084284           1.000000
reviews_per_month             -0.051965       -0.092076           0.908990
calculated_host_listings_count -0.005579        0.149947          -0.073083
availability_365               0.036689        0.092845           0.257172

                                reviews_per_month  \
price                                   -0.051965
minimum_nights                          -0.092076
number_of_reviews                        0.908990
reviews_per_month                        1.000000
calculated_host_listings_count          -0.074281
availability_365                         0.229506

                                calculated_host_listings_count  \
price                                                -0.005579
minimum_nights                                        0.149947
number_of_reviews                                    -0.073083
```

```
reviews_per_month                                 -0.074281
calculated_host_listings_count                     1.000000
availability_365                                   0.230261

                                availability_365
price                                   0.036689
minimum_nights                          0.092845
number_of_reviews                       0.257172
reviews_per_month                       0.229506
calculated_host_listings_count          0.230261
availability_365                        1.000000
```

[16]:
```python
price_correlation = correlation_matrix["price"]
filter_data = price_correlation[price_correlation > .4]
```

[17]:
```python
fig, ax = plt.subplots(figsize=(14,10))
sns.heatmap(correlation_matrix, ax=ax, linewidths=0.05,cmap="magma",annot=True)
plt.show()
```



5 Linear Regression

[18]:
```
# Linear regression matrix calculation
def normal_equation(x, y, w=None):
    if w is None:
        return np.linalg.inv(x.T.dot(x)).dot(x.T).dot(y)
    else:
        return np.linalg.inv(x.T.dot(w).dot(x)).dot(x.T).dot(w).dot(y)
```

[27]:
```
theta = normal_equation(X, Y)
theta
```

[27]:
```
array([ 1.53653049e+02,  9.71859196e-02, -7.57117577e-02, -9.63865829e+00,
       -9.04265726e-01,  1.02602014e-01])
```

6 Predict function

[29]:
```
def predict(theta, x):
    return np.dot(theta, x)
```

[32]:
```
predict(theta, X[1, :])
```

[32]:
```
181.2956996838241
```

7 RSS

[33]:
```
def rss(X, Y):
    theta = normal_equation(X, Y)
    error = X @ theta - Y
    return np.sqrt(np.sum(error**2) / Y.shape[0])
```

[34]:
```
rss(X, Y)
```

[34]:
```
241.39227025034242
```

[ ]:
```

```