Write A Data Science Blog Post

December 8, 2020

0.0.1 Information and metrics for hotels in Berlin ,Germany.

Customer satisfaction will be examined at Blaine,Germany Hotels and what are the interesting items at the room price.

Data From http://insideairbnb.com/get-the-data.html
File Name listings.csv **rename to** Berlin_Germany_Listings.csv
Date Compiled 13-10-2020

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline

        pd.set_option('display.max_rows', None)

        df_main = pd.read_csv('./Berlin_Germany_Listings.csv')

In [2]: df_main.head()
```

Out[2]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	minimum_nights	number_of_reviews
0	2015	Berlin-Mitte Value! Quiet courtyard/very central	2217	Ion	Mitte	Brunnenstr. Süd	52.53454	13.40256	Entire home/apt	61	5	134
1	3176	Fabulous Flat in great Location	3718	Britta	Pankow	Prenzlauer Berg Südwest	52.53500	13.41758	Entire home/apt	90	62	146
2	3309	BerlinSpot Schöneberg near KaDeWe	4108	Jana	Tempelhof - Schöneberg	Schöneberg-Nord	52.49885	13.34906	Private room	29	7	27
3	7071	BrightRoom with sunny greenview!	17391	BrightRoom	Pankow	Helmholtzplatz	52.54316					
4	9991	Geourgeous flat - outstanding views	33852	Philipp	Pankow	Prenzlauer Berg Südwest	52.53303					

```
3   13.41509     Private room     33               1                293
4   13.41605  Entire home/apt    180               6                  8

  last_review  reviews_per_month  calculated_host_listings_count  \
0  2020-09-26               2.43                               6
1  2020-05-27               1.06                               1
2  2019-05-31               0.31                               1
3  2020-03-31               2.15                               1
4  2020-01-04               0.13                               1

   availability_365
0               180
1               353
2               293
3                 0
4                29
```

```
In [3]: df_main.shape
```

```
Out[3]: (20227, 16)
```

```
In [4]: df_main.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20227 entries, 0 to 20226
Data columns (total 16 columns):
id                              20227 non-null int64
name                            20195 non-null object
host_id                         20227 non-null int64
host_name                       20215 non-null object
neighbourhood_group             20227 non-null object
neighbourhood                   20227 non-null object
latitude                        20227 non-null float64
longitude                       20227 non-null float64
room_type                       20227 non-null object
price                           20227 non-null int64
minimum_nights                  20227 non-null int64
number_of_reviews               20227 non-null int64
last_review                     16460 non-null object
reviews_per_month               16460 non-null float64
calculated_host_listings_count  20227 non-null int64
availability_365                20227 non-null int64
dtypes: float64(3), int64(7), object(6)
memory usage: 2.5+ MB
```

```
In [5]: df_main.describe(include='all')
```

```
Out[5]:                  id                                 name       host_id  \
        count   2.022700e+04                                20195  2.022700e+04
```

unique	NaN		19655	NaN	
top	NaN	40% off budget room with private shower		NaN	
freq	NaN		11	NaN	
mean	2.361542e+07		NaN	8.551713e+07	
std	1.372947e+07		NaN	9.821869e+07	
min	2.015000e+03		NaN	1.581000e+03	
25%	1.181564e+07		NaN	1.079583e+07	
50%	2.267455e+07		NaN	4.129505e+07	
75%	3.633792e+07		NaN	1.320462e+08	
max	4.585497e+07		NaN	3.714961e+08	

	host_name	neighbourhood_group	neighbourhood	\
count	20215	20227	20227	
unique	5541	12	138	
top	Anna	Friedrichshain-Kreuzberg	Frankfurter Allee Süd FK	
freq	152	4602	1142	
mean	NaN	NaN	NaN	
std	NaN	NaN	NaN	
min	NaN	NaN	NaN	
25%	NaN	NaN	NaN	
50%	NaN	NaN	NaN	
75%	NaN	NaN	NaN	
max	NaN	NaN	NaN	

	latitude	longitude	room_type	price	\
count	20227.000000	20227.000000	20227	20227.000000	
unique	NaN	NaN	4	NaN	
top	NaN	NaN	Entire home/apt	NaN	
freq	NaN	NaN	10971	NaN	
mean	52.510246	13.404865	NaN	67.815939	
std	0.031903	0.062069	NaN	114.235766	
min	52.340410	13.098390	NaN	0.000000	
25%	52.489590	13.369115	NaN	35.000000	
50%	52.510130	13.414540	NaN	50.000000	
75%	52.533000	13.439005	NaN	79.000000	
max	52.655980	13.757580	NaN	8000.000000	

	minimum_nights	number_of_reviews	last_review	reviews_per_month	\
count	20227.000000	20227.000000	16460	16460.000000	
unique	NaN	NaN	1843	NaN	
top	NaN	NaN	2020-10-04	NaN	
freq	NaN	NaN	251	NaN	
mean	7.990063	23.140258	NaN	0.843584	
std	30.525101	48.747977	NaN	1.218423	
min	1.000000	0.000000	NaN	0.010000	
25%	2.000000	1.000000	NaN	0.120000	
50%	3.000000	5.000000	NaN	0.360000	
75%	4.000000	20.000000	NaN	1.010000	

```
max        1124.000000         568.000000         NaN          20.630000

        calculated_host_listings_count  availability_365
count                     20227.000000      20227.000000
unique                             NaN               NaN
top                                NaN               NaN
freq                               NaN               NaN
mean                          2.823108         87.639294
std                           6.521288        128.382964
min                           1.000000          0.000000
25%                           1.000000          0.000000
50%                           1.000000          0.000000
75%                           2.000000        158.000000
max                          67.000000        365.000000
```

0.1 Questions

Question 1:- If more we go north, the lower price?

Question 2:- What is expected the price for most commonly used in the far south for every room type?

Question 3:- Increasing the number of days available per year causes price increases ?

Question 4:- when minimum nights is small then the price increases?

0.2 clean Data

select only data like the Questions and drop other columns

```
In [6]: # Drop columns
        df=df_main.drop(['name' , 'host_id' , 'host_name', 'neighbourhood_group','neighbourhood'
```

drop rows is null values

```
In [7]: df=df.dropna()

In [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20227 entries, 0 to 20226
Data columns (total 6 columns):
id                  20227 non-null int64
latitude            20227 non-null float64
room_type           20227 non-null object
price               20227 non-null int64
minimum_nights      20227 non-null int64
availability_365    20227 non-null int64
```

```
dtypes: float64(1), int64(4), object(1)
memory usage: 1.1+ MB
```

Drop rows is number of reviews equal 0

```
In [9]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 20227 entries, 0 to 20226
Data columns (total 6 columns):
id                  20227 non-null int64
latitude            20227 non-null float64
room_type           20227 non-null object
price               20227 non-null int64
minimum_nights      20227 non-null int64
availability_365    20227 non-null int64
dtypes: float64(1), int64(4), object(1)
memory usage: 1.1+ MB
```

```
In [10]: df.describe(include='all')

Out[10]:                   id      latitude        room_type          price  \
         count   2.022700e+04  20227.000000            20227  20227.000000
         unique           NaN           NaN                4           NaN
         top              NaN           NaN  Entire home/apt           NaN
         freq             NaN           NaN            10971           NaN
         mean    2.361542e+07     52.510246              NaN     67.815939
         std     1.372947e+07      0.031903              NaN    114.235766
         min     2.015000e+03     52.340410              NaN      0.000000
         25%     1.181564e+07     52.489590              NaN     35.000000
         50%     2.267455e+07     52.510130              NaN     50.000000
         75%     3.633792e+07     52.533000              NaN     79.000000
         max     4.585497e+07     52.655980              NaN   8000.000000

                 minimum_nights  availability_365
         count     20227.000000      20227.000000
         unique             NaN               NaN
         top                NaN               NaN
         freq               NaN               NaN
         mean          7.990063         87.639294
         std          30.525101        128.382964
         min           1.000000          0.000000
         25%           2.000000          0.000000
         50%           3.000000          0.000000
         75%           4.000000        158.000000
         max        1124.000000        365.000000
```

```
In [11]: df.groupby(['room_type']).count()
```

Out[11]:

room_type	id	latitude	price	minimum_nights	availability_365
Entire home/apt	10971	10971	10971	10971	10971
Hotel room	230	230	230	230	230
Private room	8747	8747	8747	8747	8747
Shared room	279	279	279	279	279

```
In [12]: df['room_type']=df['room_type'].str.replace(' ','_')

In [13]: df['room_type'].replace('Entire_home/apt','Entire_home',inplace=True)

In [14]: df.groupby(['room_type']).count()
```

Out[14]:

room_type	id	latitude	price	minimum_nights	availability_365
Entire_home	10971	10971	10971	10971	10971
Hotel_room	230	230	230	230	230
Private_room	8747	8747	8747	8747	8747
Shared_room	279	279	279	279	279

```
In [15]: # convert room type to columns
         cat_df = df.select_dtypes(include=['object'])
         cat_cols = cat_df.columns
         for col in  cat_cols:
             df = pd.concat([df.drop(col, axis=1), pd.get_dummies(df[col], prefix=col, prefix_se

In [16]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 20227 entries, 0 to 20226
Data columns (total 10 columns):
id                          20227 non-null int64
latitude                    20227 non-null float64
price                       20227 non-null int64
minimum_nights              20227 non-null int64
availability_365            20227 non-null int64
room_type_Entire_home       20227 non-null uint8
room_type_Hotel_room        20227 non-null uint8
room_type_Private_room      20227 non-null uint8
room_type_Shared_room       20227 non-null uint8
room_type_nan               20227 non-null uint8
dtypes: float64(1), int64(4), uint8(5)
memory usage: 1.0 MB


In [17]: # Lift the decimalfor latitude
         df['latitude']=(df['latitude']-52)*100
```

```
In [18]: df['latitude'].head()

Out[18]: 0    53.454
         1    53.500
         2    49.885
         3    54.316
         4    53.303
         Name: latitude, dtype: float64

In [19]: # convert latitude to int
         df=df.astype(int)

In [20]: df.head()

Out[20]:      id  latitude  price  minimum_nights  availability_365  \
         0  2015        53     61               5               180
         1  3176        53     90              62               353
         2  3309        49     29               7               293
         3  7071        54     33               1                 0
         4  9991        53    180               6                29

            room_type_Entire_home  room_type_Hotel_room  room_type_Private_room  \
         0                      1                     0                       0
         1                      1                     0                       0
         2                      0                     0                       1
         3                      0                     0                       1
         4                      1                     0                       0

            room_type_Shared_room  room_type_nan
         0                      0              0
         1                      0              0
         2                      0              0
         3                      0              0
         4                      0              0

In [21]: df['latitude'].head()

Out[21]: 0    53
         1    53
         2    49
         3    54
         4    53
         Name: latitude, dtype: int64

In [22]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 20227 entries, 0 to 20226
Data columns (total 10 columns):
```

```
id                          20227 non-null int64
latitude                    20227 non-null int64
price                       20227 non-null int64
minimum_nights              20227 non-null int64
availability_365            20227 non-null int64
room_type_Entire_home       20227 non-null int64
room_type_Hotel_room        20227 non-null int64
room_type_Private_room      20227 non-null int64
room_type_Shared_room       20227 non-null int64
room_type_nan               20227 non-null int64
dtypes: int64(10)
memory usage: 1.7 MB
```

0.3 Answer Question

Question 1:- If more we go south, the lower the price?

In [23]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
```

In [24]:
```
#Split into explanatory and response variables
X = df[['latitude', 'minimum_nights','availability_365','room_type_Entire_home','room_t
y = df['price']

#Split into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = .30, random_state

lm_model = LinearRegression(normalize=True) # Instantiate
lm_model.fit(X_train, y_train) #Fit

#Predict and score the model
y_test_preds = lm_model.predict(X_test)

"The r-squared score for the model using only quantitative variables was {} on {} value
```

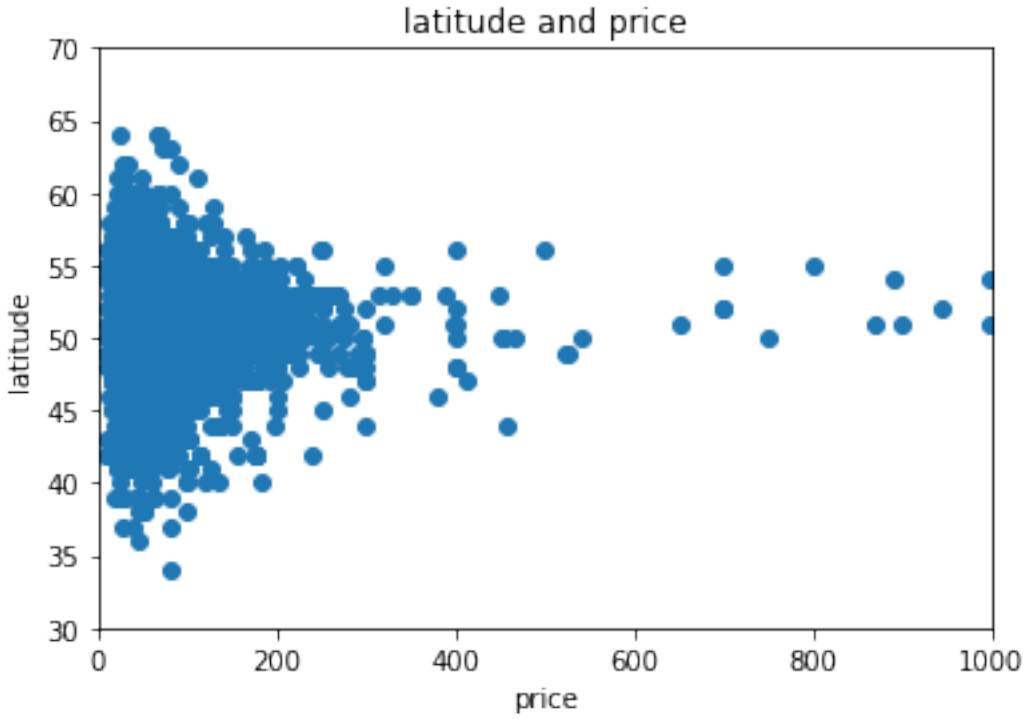
Out[24]:
```
'The r-squared score for the model using only quantitative variables was 0.044210269478
```

In [25]:
```
X_test['price']=y_test
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py: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: http://pandas.pydata.org/pandas-docs/stable/indexing.html#
  """Entry point for launching an IPython kernel.
```

In [26]:
```
X_test['latitude'].describe(include='all')
```

```
Out[26]: count    6069.000000
         mean       50.592190
         std         3.181286
         min        34.000000
         25%        49.000000
         50%        51.000000
         75%        53.000000
         max        64.000000
         Name: latitude, dtype: float64

In [27]: # convert latitude to int
         X_test=X_test.astype(int)

In [28]: plt.scatter( X_test['price'],X_test['latitude'])
         plt.axis([0,1000, 30,70]) # set axis
         plt.title("latitude and price")
         plt.ylabel("latitude")
         plt.xlabel("price")
         plt.show()
```



Answer 1:- Yes, If more we go north, the lower price and Hotel Count.

Question 2:- What is expected the price for most commonly used in the far south for every room type?

```
In [29]: X_test.head()
```

Out[29]:

	latitude	minimum_nights	availability_365	room_type_Entire_home	\
15376	49	2	0	1	
10404	51	2	168	1	
20026	57	1	179	0	
2043	49	3	0	1	
11508	48	4	0	0	

	room_type_Hotel_room	room_type_Private_room	room_type_Shared_room	\
15376	0	0	0	
10404	0	0	0	
20026	0	1	0	
2043	0	0	0	
11508	0	1	0	

	room_type_nan	price
15376	0	90
10404	0	174
20026	0	33
2043	0	55
11508	0	48

```
In [30]: df_X_where=X_test.query('latitude<= 45')

In [31]: df_X_where.head()
```

Out[31]:

	latitude	minimum_nights	availability_365	room_type_Entire_home	\
119	44	2	320	1	
7430	43	2	36	0	
3571	40	2	276	0	
4505	44	1	0	1	
15388	43	21	364	1	

	room_type_Hotel_room	room_type_Private_room	room_type_Shared_room	\
119	0	0	0	
7430	0	1	0	
3571	0	1	0	
4505	0	0	0	
15388	0	0	0	

	room_type_nan	price
119	0	52
7430	0	24
3571	0	23
4505	0	95
15388	0	28

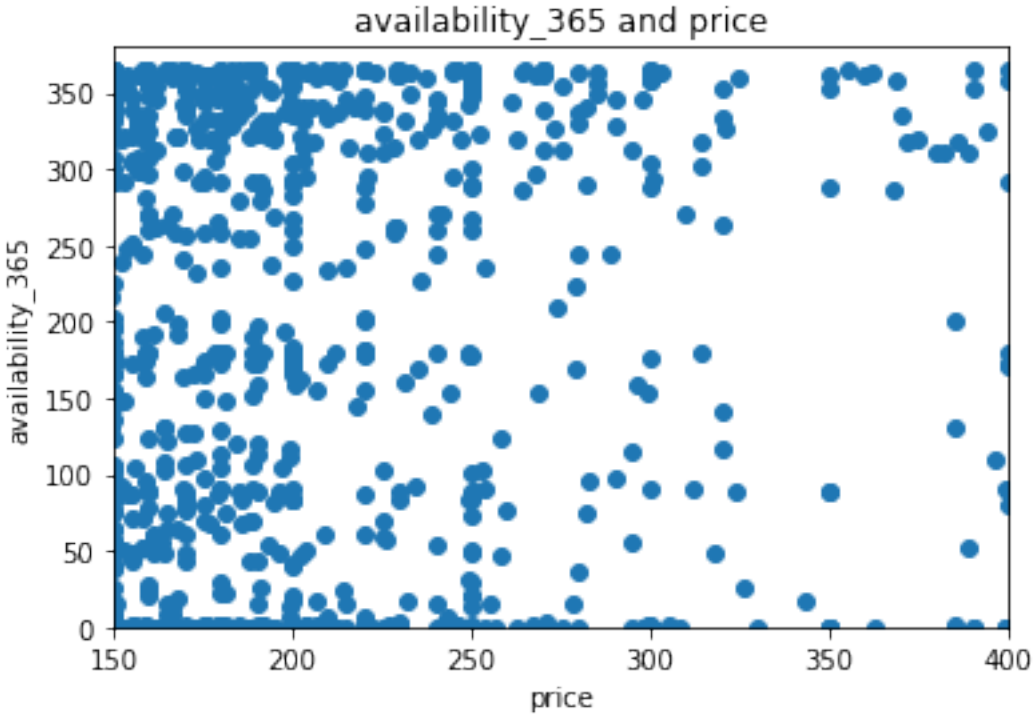
```
In [32]: df_X_where[df_X_where['room_type_Entire_home']==1]['price'].std()
```

```
Out[32]: 54.139040574115405

In [33]: df_X_where[df_X_where['room_type_Hotel_room']==1]['price'].std()

Out[33]: nan

In [34]: df_X_where[df_X_where['room_type_Private_room']==1]['price'].std()

Out[34]: 30.231309678773144

In [35]: df_X_where[df_X_where['room_type_Shared_room']==1]['price'].std()

Out[35]: 35.0
```

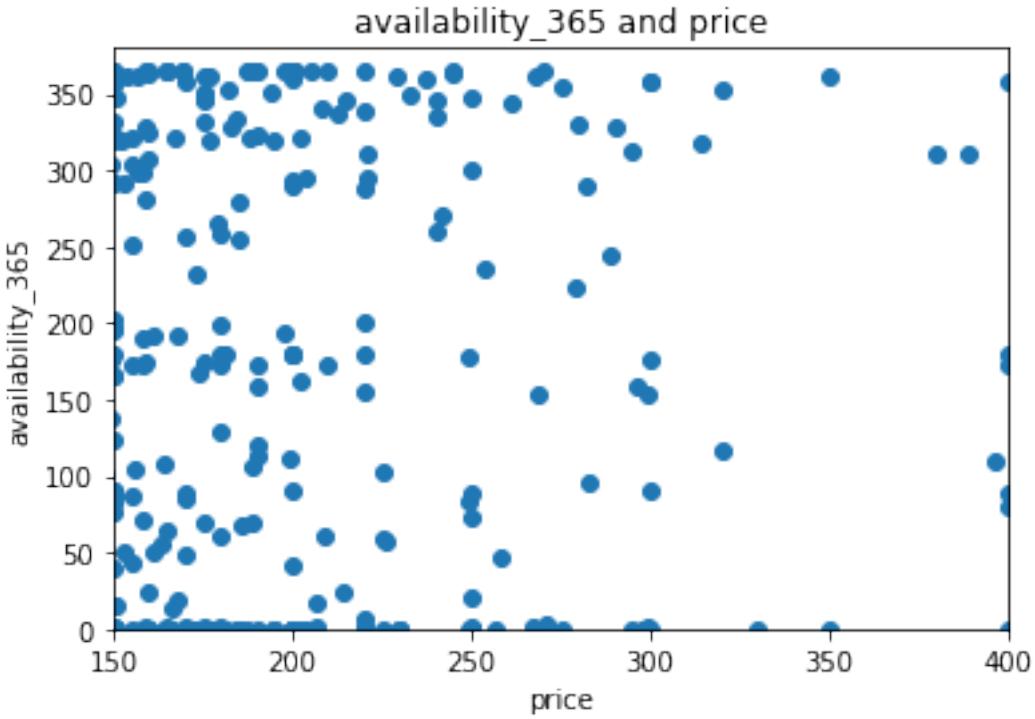
Answer 2:- the expected price for most commonly used in the far south is 61.3 Euro from Entire home and 16.2 Euro from Shared room.

Question 3:- Increasing the number of days available per year causes price increases ?

```
In [36]: plt.scatter( df['price'],df['availability_365'])
         plt.axis([150,400, 0,380]) # set axis
         plt.title("availability_365 and price")
         plt.ylabel("availability_365")
         plt.xlabel("price")
         plt.show()
```



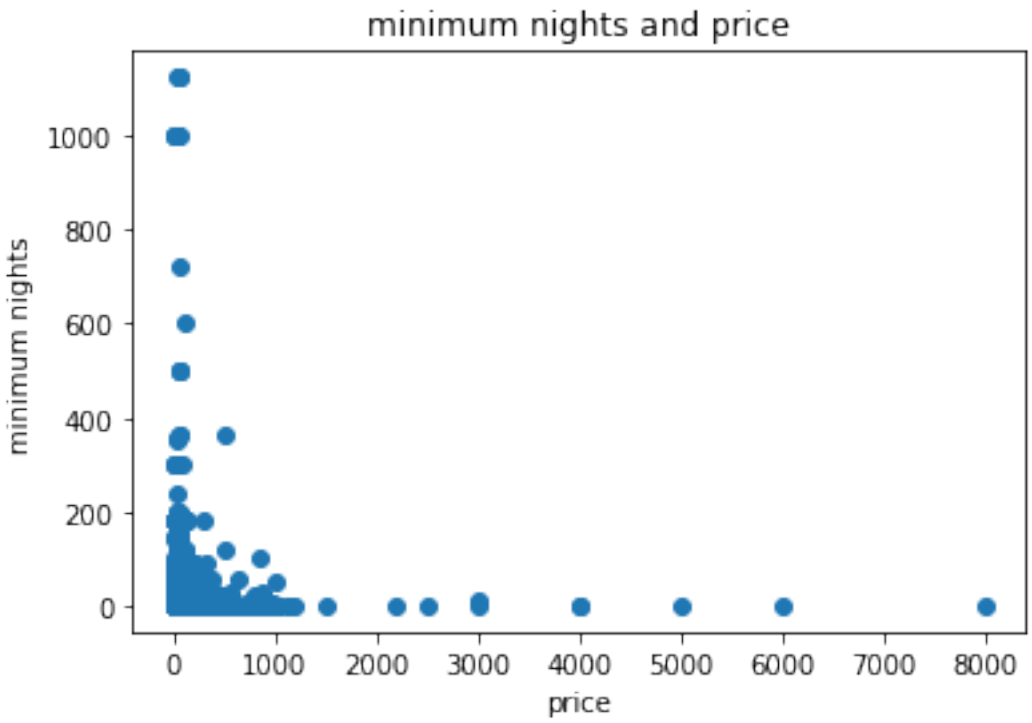
```
In [37]: plt.scatter( X_test['price'],X_test['availability_365'])
         plt.axis([150,400, 0,380]) # set axis
         plt.title("availability_365 and price")
         plt.ylabel("availability_365")
         plt.xlabel("price")
         plt.show()
```



Answer 3:- Yes, Increasing the number of days available per year causes price increases .

Question 4:- when minimum nights is small then the price increases?

```
In [38]: plt.scatter( df['price'],df['minimum_nights'])

         plt.title("minimum nights and price")
         plt.ylabel("minimum nights")
         plt.xlabel("price")
         plt.show()
```



Answer 4:- No, when the minimum nights is small, the price move to decreases.

0.3.1 Summary

-When we go north, the lower price and Hotel Count.

-The expected price for most commonly used in the far south is 61.3 Euro from Entire home and 16.2 Euro from Shared room.

-Increasing the number of days available per year causes price increases .

-Not effect when the minimum nights is small the price not decreases.

In [ ]: