In [12]:
```python
import pandas as pd
import seaborn as sns
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

In [14]:
```python
airbnb = pd.read_csv("Datasets/seattle_01.csv")
airbnb = airbnb.drop(["Unnamed: 0"] ,axis = 1)
```

Cleaning the data

In [3]:
```python
airbnb.isnull().sum()
```

Out[3]:
```
room_id                    0
host_id                    0
room_type                  0
address                    0
reviews                    0
overall_satisfaction    1473
accommodates               0
bedrooms                   0
bathrooms                  2
price                      0
last_modified              0
latitude                   0
longitude                  0
location                   0
name                       0
currency                   0
rate_type                  0
dtype: int64
```

In [4]:
```python
#Filling numeric missing values
airbnb = airbnb.fillna(airbnb.mean())
airbnb.isnull().sum()
```

Out[4]:
```
room_id                 0
host_id                 0
room_type               0
address                 0
reviews                 0
overall_satisfaction    0
accommodates            0
bedrooms                0
bathrooms               0
price                   0
last_modified           0
latitude                0
longitude               0
location                0
name                    0
currency                0
rate_type               0
dtype: int64
```

In [5]:
```python
#Filling categorical missing values
airbnb=airbnb.fillna(method="ffill")
airbnb.isnull().sum()
```

Out[5]:
```
room_id                 0
host_id                 0
room_type               0
address                 0
reviews                 0
overall_satisfaction    0
accommodates            0
bedrooms                0
bathrooms               0
price                   0
last_modified           0
latitude                0
longitude               0
location                0
name                    0
currency                0
rate_type               0
dtype: int64
```

In [6]:
```python
airbnb.to_csv("airbnb_clean.csv")
```

All missing values have been filled

Normalizing the data

In [7]:
```python
airbnb.dtypes
```

Out[7]:
```
room_id                   int64
host_id                   int64
room_type                object
address                  object
reviews                   int64
overall_satisfaction    float64
accommodates              int64
bedrooms                float64
bathrooms               float64
price                     int64
last_modified            object
latitude                float64
longitude               float64
location                 object
name                     object
currency                 object
rate_type                object
dtype: object
```

In [8]:
```python
# Normalizing only the continuous values
airbnb_continuous = airbnb.drop(["room_type" , "rate_type" , "currency" ,"address","last_modified","location","name"], axis = 1)
airbnb_continuous =  (airbnb_continuous - airbnb_continuous.mean())/airbnb_continuous.std()
airbnb_normalized = airbnb_continuous
# Adding the categorical columns back
airbnb_normalized["room_type"] = airbnb["room_type"]
airbnb_normalized["rate_type"] = airbnb["rate_type"]
airbnb_normalized["currency"] = airbnb["currency"]
airbnb_normalized["name"] = airbnb["name"]
airbnb_normalized["address"] = airbnb["address"]
airbnb_normalized["last_modified"] = airbnb["last_modified"]
airbnb_normalized["location"] = airbnb["location"]
```

1. Why is normalization important?

Normalization is important since different columns may have different range of values. It helps ensure that all the columns fall within the same range of values and are dispered equally

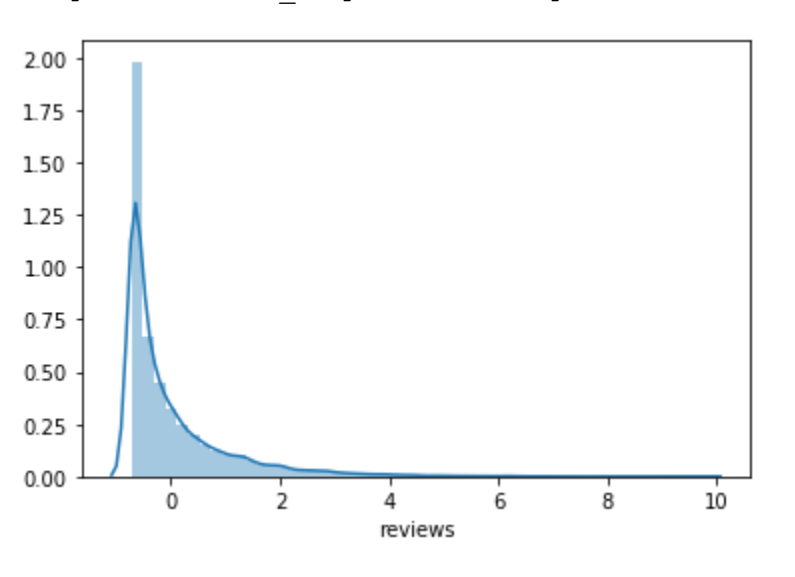
1. How does it affect the dataset?

It makes all the values unitless and easy to compare

1. Different graphs to check whether data is normal :

In [9]:
```python
sns.distplot(airbnb_normalized["reviews"])
```

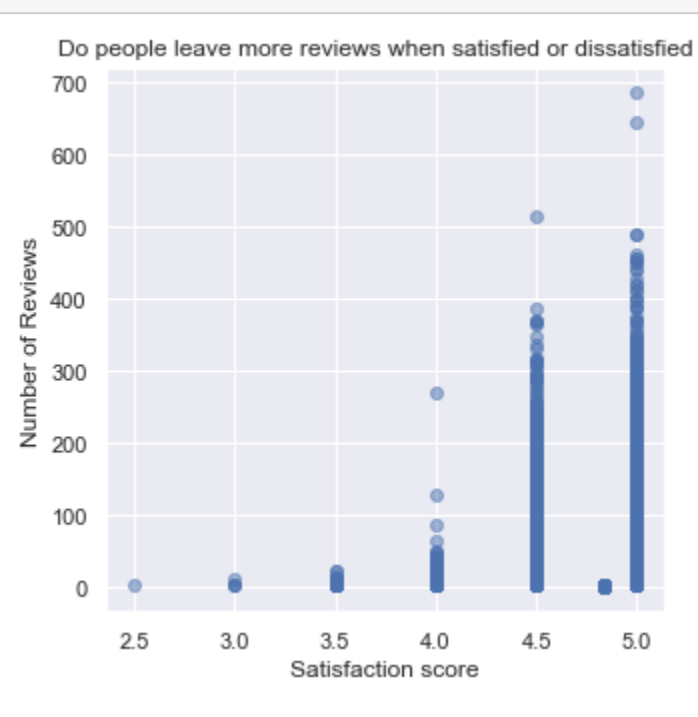
Out[9]: <matplotlib.axes._subplots.AxesSubplot at 0x1a16bb3128>



Visualizations

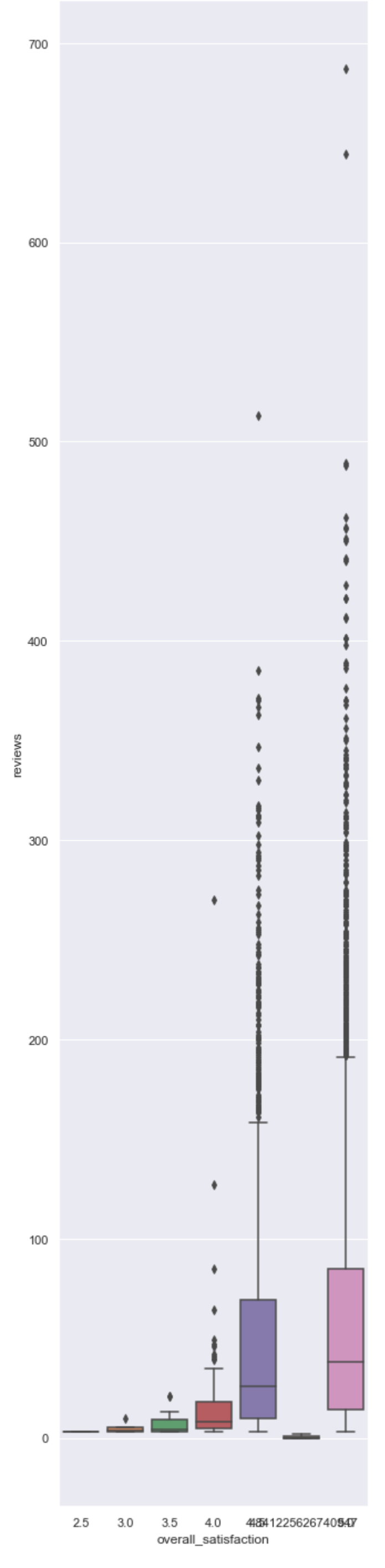
In [10]:
```python
#visualizations with cleaned data

import matplotlib.pyplot as plt
sns.set(rc={'figure.figsize':(5,5)})
#do people leave more reviews when satisfied or dissatisfied
plt.scatter(airbnb["overall_satisfaction"], airbnb["reviews"], alpha=0.5)
plt.title('Do people leave more reviews when satisfied or dissatisfied ?')
plt.xlabel('Satisfaction score')
plt.ylabel('Number of Reviews')
plt.show()
```

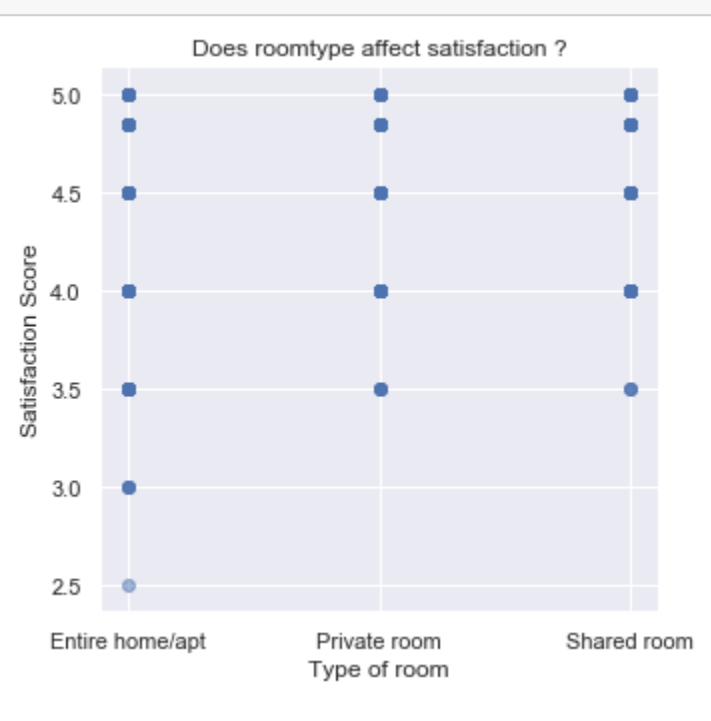


In [11]:
```python
sns.set(rc={'figure.figsize':(5,25.00)})
sns.boxplot(airbnb["overall_satisfaction"], airbnb["reviews"])
```

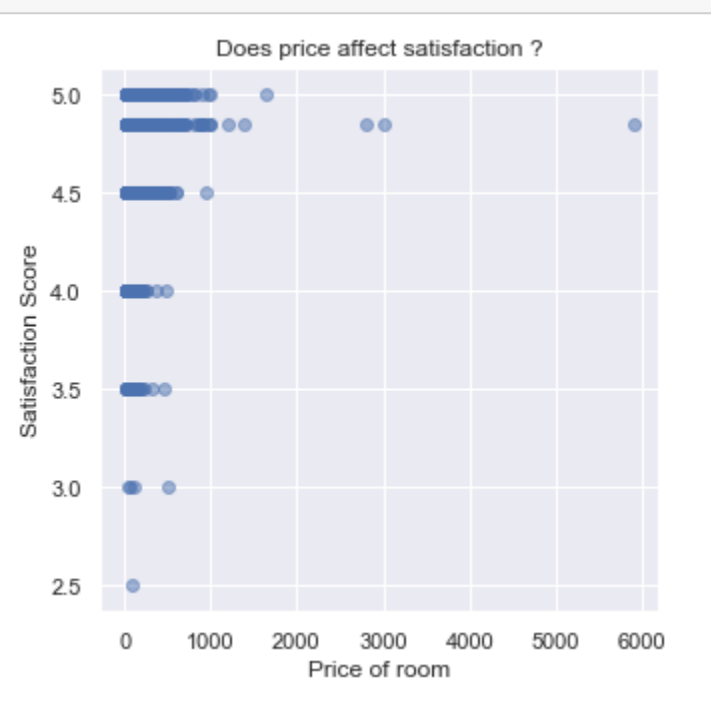
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1a16c78390>



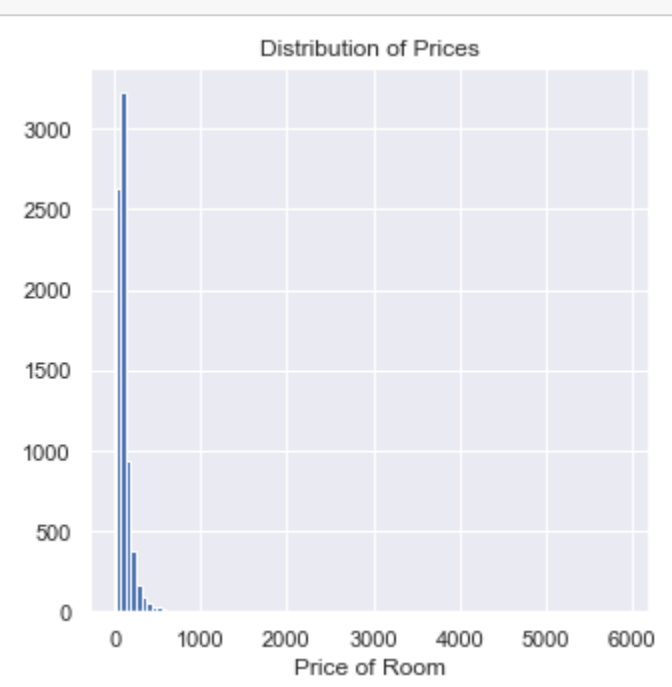
In [12]:
```python
#does roomtype affect satisfaction
sns.set(rc={'figure.figsize':(5,5)})
plt.scatter(airbnb["room_type"], airbnb["overall_satisfaction"], alpha=0.5)
plt.title('Does roomtype affect satisfaction ?')
plt.xlabel('Type of room')
plt.ylabel('Satisfaction Score')
plt.show()
```



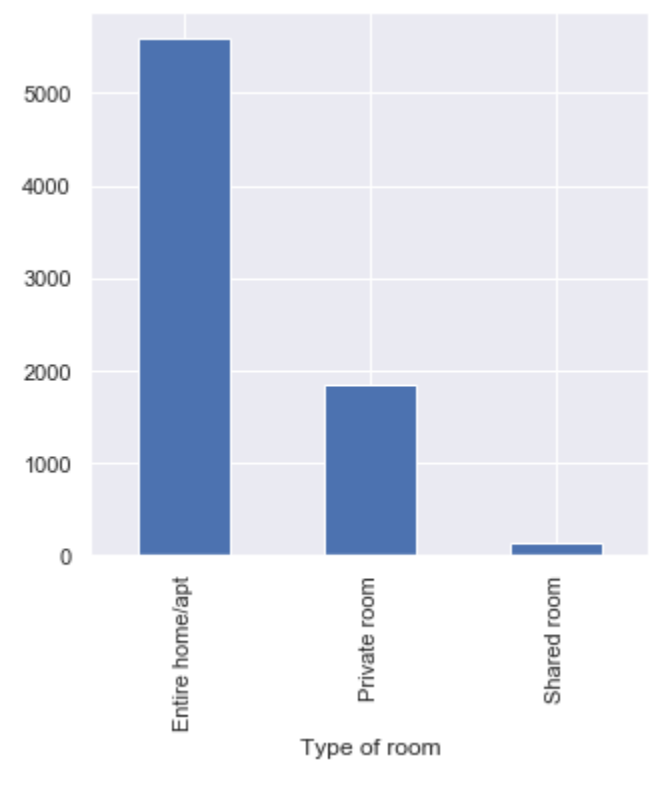
In [13]:
```python
#does price affect satisfaction
sns.set(rc={'figure.figsize':(5,5)})
plt.scatter(airbnb["price"], airbnb["overall_satisfaction"], alpha=0.5)
plt.title('Does price affect satisfaction ?')
plt.xlabel('Price of room')
plt.ylabel('Satisfaction Score')
plt.show()
```



In [19]:
```python
#what is the dirtribution of prices
sns.set(rc={'figure.figsize':(5,5)})
plt.hist(airbnb["price"],bins=100)
plt.xlabel('Price of Room')
plt.title('Distribution of Prices')
plt.show()
```

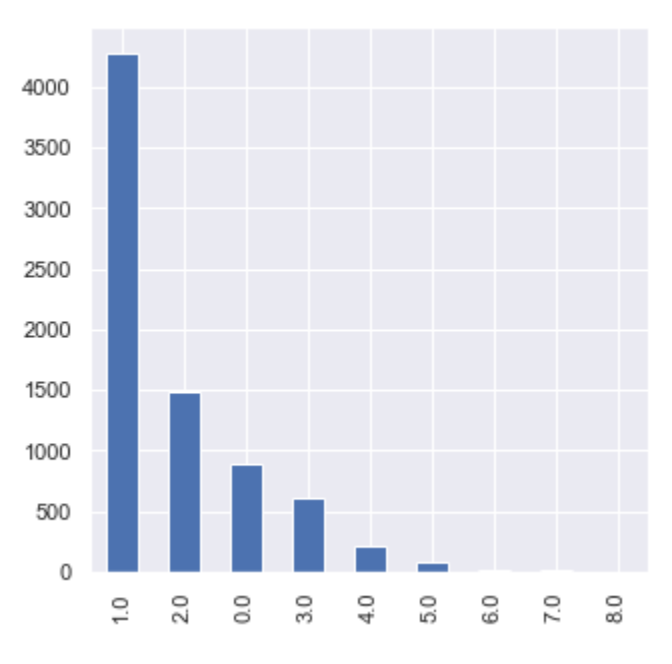


In [15]:
```python
#How many kinds of houses are up for rental
airbnb['room_type'].value_counts().plot(kind='bar')
plt.xlabel("Type of room")
plt.show()
```



In [16]:
```python
airbnb['bedrooms'].value_counts().plot(kind='bar')
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x1a18051ef0>



Hypothesis

In [20]:
```python
import pingouin as pg
pg.corr(x=airbnb["price"], y=airbnb["overall_satisfaction"])
```

Out[20]:

	n	r	CI95%	r2	adj_r2	p-val	BF10	power
pearson	7576	0.042	[0.02, 0.06]	0.002	0.002	0.000253	11.613	0.955

In [21]:
```python
pg.corr(x=airbnb["price"], y=airbnb["bedrooms"])
```

Out[21]:

	n	r	CI95%	r2	adj_r2	p-val	BF10	power
pearson	7576	0.462	[0.44, 0.48]	0.214	0.214	0.0	inf	1.0

In [22]:
```python
pg.corr(x=airbnb["bedrooms"], y=airbnb["overall_satisfaction"])
```

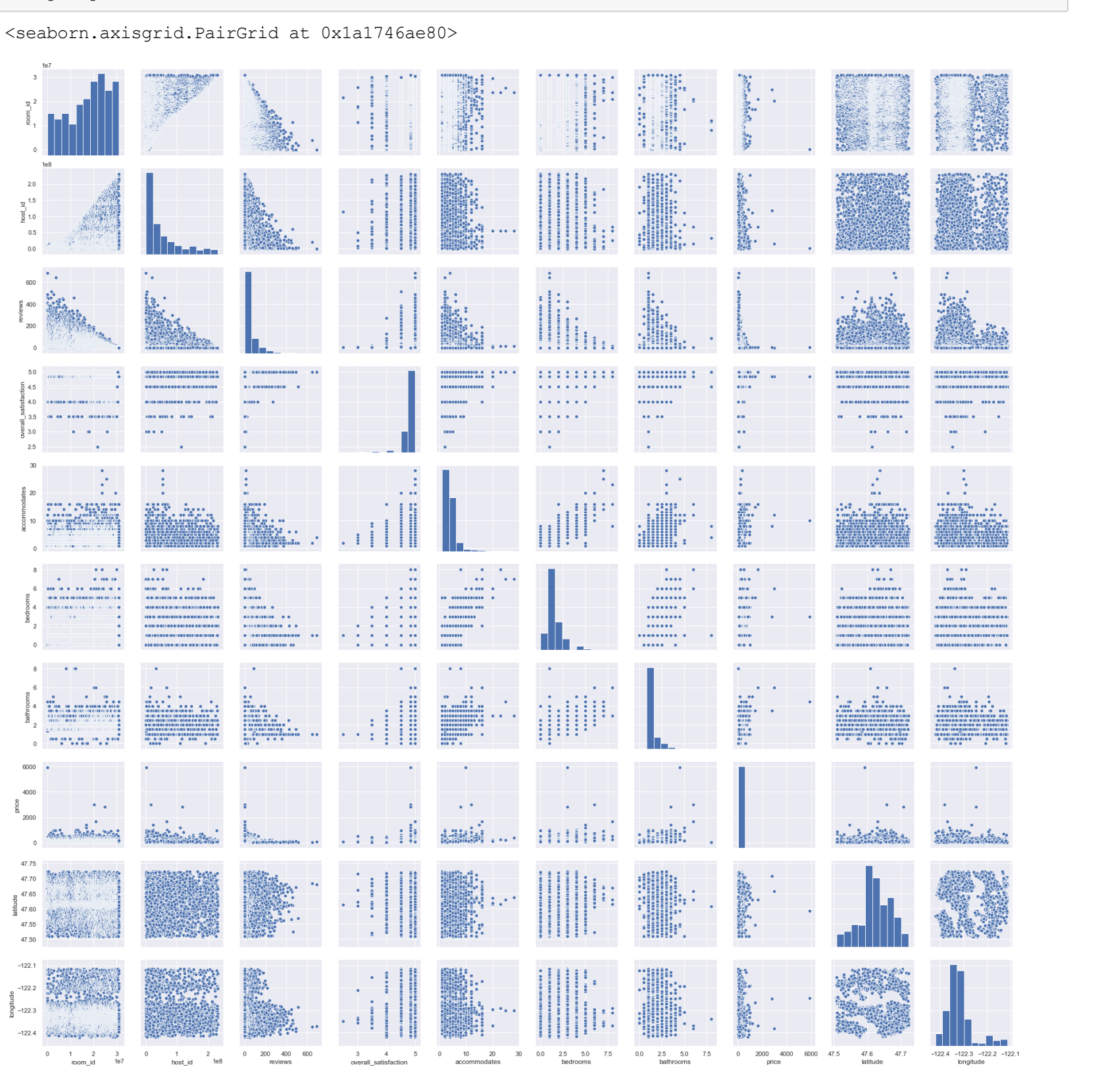
Out[22]:

	n	r	CI95%	r2	adj_r2	p-val	BF10	power
pearson	7576	0.029	[0.01, 0.05]	0.001	0.001	0.012622	0.323	0.704

Corelation Matrix:

In [18]:
```python
sns.pairplot(airbnb)
```

Out[18]: <seaborn.axisgrid.PairGrid at 0x1a1746ae80>



In [32]:
```python
continuous = airbnb.drop(["room_type" , "rate_type" , "currency" ,"address","last_modified","location","name"], axis = 1)
np.cov(continuous)
```

Out[32]:
```
array([[1.03493184e+06, 1.61893420e+06, 8.92283901e+06, ...,
        5.81926898e+10, 1.70602587e+10, 1.70603018e+10],
       [1.61893420e+06, 2.55623891e+06, 1.48752487e+07, ...,
        9.68731541e+10, 2.72101190e+10, 2.72101796e+10],
       [8.92283901e+06, 1.48752487e+07, 1.26164359e+08, ...,
        8.13635690e+11, 1.70245279e+11, 1.70245278e+11],
       ...,
       [5.81926898e+10, 9.68731541e+10, 8.13635690e+11, ...,
        5.24842100e+15, 1.10649326e+15, 1.10649332e+15],
       [1.70602587e+10, 2.72101190e+10, 1.70245279e+11, ...,
        1.10649326e+15, 2.93472947e+14, 2.93473486e+14],
       [1.70603018e+10, 2.72101796e+10, 1.70245278e+11, ...,
        1.10649332e+15, 2.93473486e+14, 2.93474024e+14]])
```

In [42]:
```python
f = plt.figure(figsize=(12, 12))
plt.matshow(continuous.corr(), fignum=f.number)
plt.xticks(range(continuous.shape[1]), continuous.columns, fontsize=14, rotation=45)
plt.yticks(range(continuous.shape[1]), continuous.columns, fontsize=14)
cb = plt.colorbar()
cb.ax.tick_params(labelsize=14)
plt.title('Correlation Matrix', fontsize=16);
```