Project 1

Author: Zeyao Wang

In [1]:
```
import pandas as pd
import numpy as np
from time import strptime
import matplotlib.pyplot as plt

pd.set_option("display.max_columns", None)
```

In [2]:
```
# Read data
calendar = pd.read_csv('data/Seattle/calendar.csv')
listings = pd.read_csv('data/Seattle/listings.csv')
reviews = pd.read_csv('data/Seattle/reviews.csv')

reviews = reviews.rename(columns = {'id':'review_id'})
reviews.head(1)
```

Out[2]:

	listing_id	review_id	date	reviewer_id	reviewer_name	comments
0	7202016	38917982	2015-07-19	28943674	Bianca	Cute and cozy place. Perfect location to every...

Data Cleaning

- check missing data (quantitative and categorical variables)

In [3]:
```
# Check Missing data
missing = (listings.isnull().sum() / listings.shape[0]).sort_values(ascending=Fal
missing = missing[missing > 0]
missing
```

Out[3]:
```
license                        1.000000
square_feet                    0.974594
monthly_price                  0.602672
security_deposit               0.511262
weekly_price                   0.473808
notes                          0.420639
neighborhood_overview          0.270299
cleaning_fee                   0.269775
transit                        0.244631
host_about                     0.224987
host_acceptance_rate           0.202462
review_scores_accuracy         0.172342
review_scores_checkin          0.172342
review_scores_value            0.171818
review_scores_location         0.171556
review_scores_cleanliness      0.171032
review_scores_communication    0.170508
review_scores_rating           0.169460
reviews_per_month              0.164222
```

- Drop missing data columns with more than 60% missing values:

In [4]:
```
# drop missing data columns with more than 60% NA
listings2 = listings.drop(['license', 'square_feet'], axis=1)
```

In [5]:
```
# Check Quantitative and Quanlitative Columns
quant = []
quali = []

for i in listings2.columns:
    if listings2.dtypes[i] != 'object':
        quant.append(i)
    else:
        quali.append(i)
```

In [6]:
```
# Check missing qualitative columns
missing_quali = listings2[quali].isnull().sum().sort_values(ascending=False)
missing_quali = missing_quali[missing_quali > 0]
missing_quali
```

Out[6]:
```
notes                     1606
neighborhood_overview     1032
transit                    934
host_about                 859
last_review                627
first_review               627
space                      569
host_response_time         523
neighbourhood              416
xl_picture_url             320
thumbnail_url              320
medium_url                 320
host_neighbourhood         300
summary                    177
host_location                8
zipcode                      7
host_is_superhost            2
host_thumbnail_url           2
host_has_profile_pic         2
host_since                   2
host_identity_verified       2
host_name                    2
host_picture_url             2
property_type                1
dtype: int64
```

In [7]:
```
# Fill NA in qualitative columns
for i in missing_quali.index:
    listings2[i] = listings2[i].fillna('None')
```

In [8]:
```
# Check again
missing_quali = listings2[quali].isnull().sum().sort_values(ascending=False)
missing_quali = missing_quali[missing_quali > 0]
missing_quali.head()
```

Out[8]:
```
Series([], dtype: int64)
```

In [9]:
```
# Check missing quantitative columns
missing_quant = (listings2[quant].isnull().sum() / listings2.shape[0]).sort_value
missing_quant = missing_quant[missing_quant > 0]
missing_quant
```

Out[9]:
```
monthly_price                   0.602672
security_deposit                0.511262
weekly_price                    0.473808
cleaning_fee                    0.269775
host_acceptance_rate            0.202462
review_scores_checkin           0.172342
review_scores_accuracy          0.172342
review_scores_value             0.171818
review_scores_location          0.171556
review_scores_cleanliness       0.171032
review_scores_communication     0.170508
review_scores_rating            0.169460
reviews_per_month               0.164222
host_response_rate              0.136983
bathrooms                       0.004191
bedrooms                        0.001572
host_listings_count             0.000524
host_total_listings_count       0.000524
beds                            0.000262
dtype: float64
```

Need to check details to figure out how to deal with missing value for quantitative columns

In [10]:
```
# Fill quant missing value with its median
for i in missing_quant.index:
    listings2[i] = listings2[i].fillna(listings2[i].median())
```

In [11]:
```
# Check again
missing_quant = (listings2[quant].isnull().sum() / listings2.shape[0] * 100).sort_
missing_quant = missing_quant[missing_quant > 0]
missing_quant
```

Out[11]:
```
Series([], dtype: float64)
```

In [12]:
```
df = listings2
```

In [13]:
```
listings2.groupby(['id'])['last_scraped'].size().sort_values(ascending = False).h
```

Out[13]:
```
id
10340165    1
4104442     1
4126284     1
4125779     1
4122325     1
Name: last_scraped, dtype: int64
```

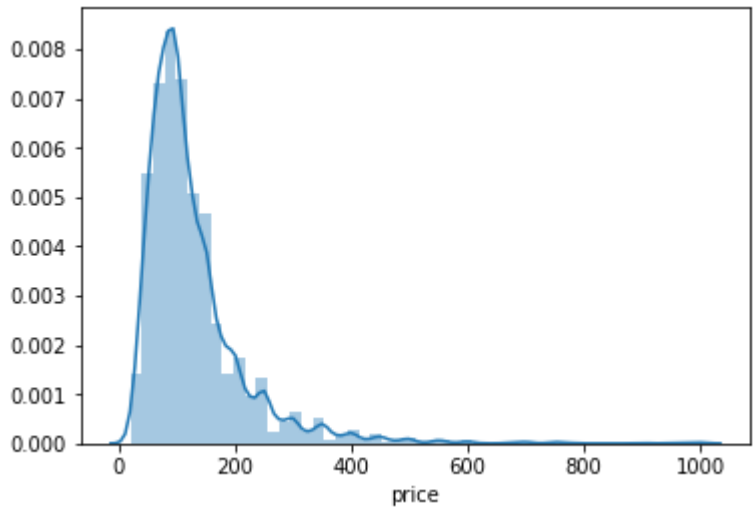
Data Prepare

- Check price value

In [14]:
```
import seaborn as sns
```

In [15]:
```
sns.distplot((df.price))
```

Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x201d4b79c18>
```

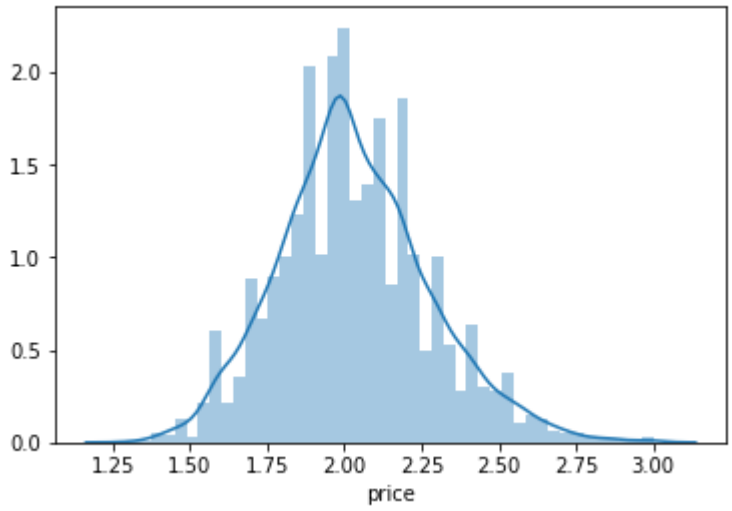


In [16]:
```
#skewness and kurtosis
print("Skewness: %f" % df.price.skew())
print("Kurtosis: %f" % df.price.kurt())
```

```
Skewness: 3.113123
Kurtosis: 16.617132
```

In [17]:
```
sns.distplot(np.log10(df.price))
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x201d622ae10>
```



After checking orginal price and log(price) distribution, log(price) shows in more normal distribution. Then use log(price) in the further analysis.

In [18]:
```
#skewness and kurtosis
print("Skewness: %f" % np.log10(df.price).skew())
print("Kurtosis: %f" % np.log10(df.price).kurt())
```

```
Skewness: 0.372414
Kurtosis: 0.371624
```

Skew and Kurtosis is between -0.5 and 0.5, the log(price) distribution is fairly symmetrical.

In [19]:
```
df2 = df[['id', 'name', 'price', 'host_response_rate', 'host_acceptance_rate', 'h
    'latitude', 'longitude', 'property_type', 'accommodates',
       'bathrooms', 'bedrooms', 'beds', 'review_scores_rating']]
df2.head(1)
```

Out[19]:

	id	name	price	host_response_rate	host_acceptance_rate	host_is_superhost	host_has
0	241032	Stylish Queen Anne Apartment	85	0.96	1.0	f	

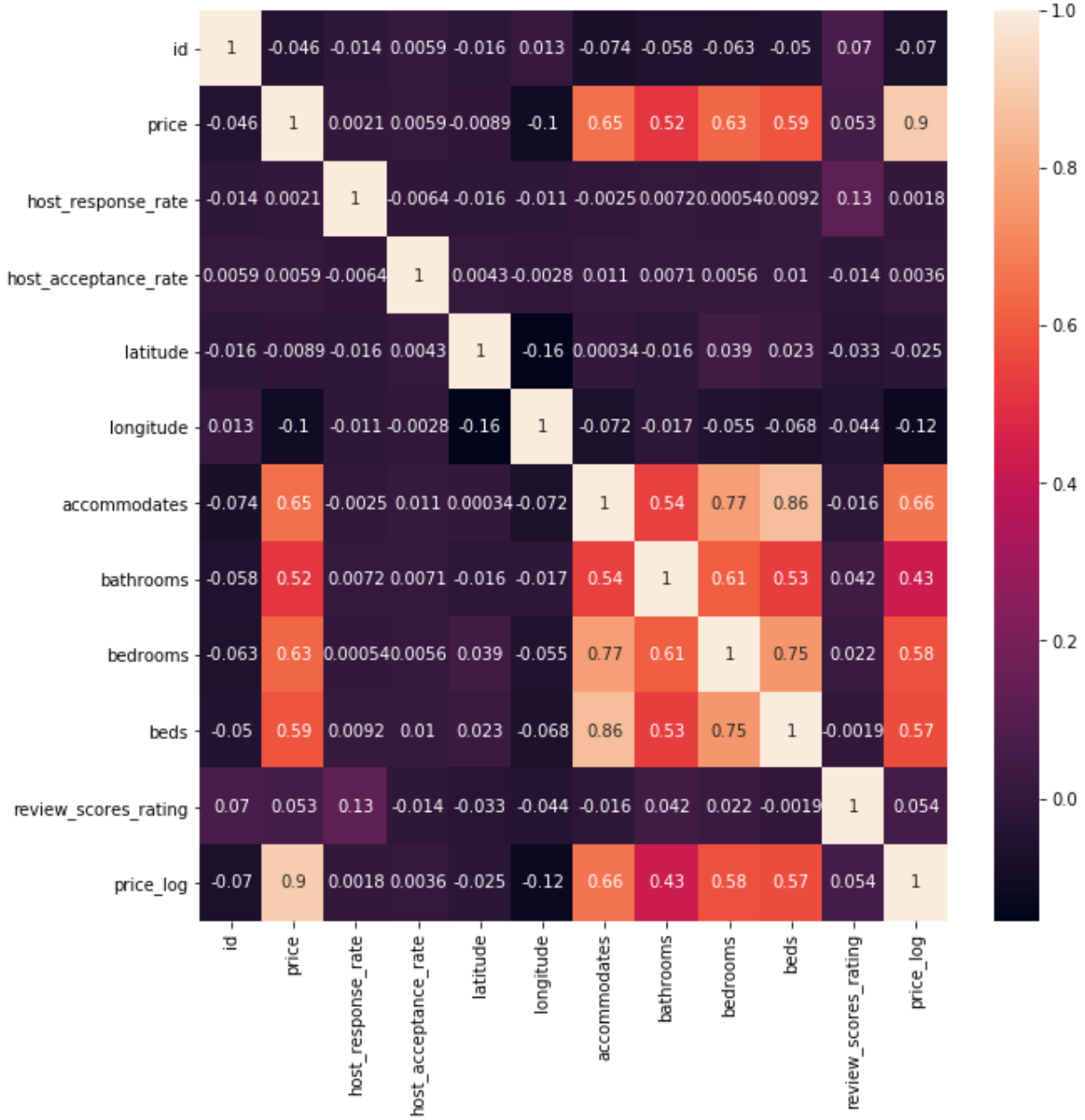
In [20]:
```
df2['price_log'] = np.log10(df2.price)
```

```
C:\Anaconda3\lib\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#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
```

- Check the correlation between price with other features

In [21]:
```
fig, ax = plt.subplots(figsize=(10, 10))
sns.heatmap(df2.corr(), annot = True)
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x201d0d3bc88>



Price shows the positive correlation with accommondates indicators, bathroom numbers, bedroom numbers, beds numbers. Meanwhile, accommodats have strong positive correlation with bathrooms, bedrooms.

Model

In [22]:
```
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from sklearn import linear_model
```

```
C:\Anaconda3\lib\site-packages\statsmodels\compat\pandas.py:56: FutureWarning:
The pandas.core.datetools module is deprecated and will be removed in a future
version. Please use the pandas.tseries module instead.
  from pandas.core import datetools
```

In [23]:
```
# Change to dummy variables
df2['superhost'] = np.where(df2['host_is_superhost'] == "t", 1, 0)
df2['profile_pic'] = np.where(df2['host_has_profile_pic'] == "t", 1, 0)
```

```
C:\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: 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#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)

C:\Anaconda3\lib\site-packages\ipykernel_launcher.py: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: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
  This is separate from the ipykernel package so we can avoid doing imports until
```

In [24]:
```
df2.columns
```

Out[24]:
```
Index(['id', 'name', 'price', 'host_response_rate', 'host_acceptance_rate',
       'host_is_superhost', 'host_has_profile_pic', 'latitude', 'longitude',
       'property_type', 'accommodates', 'bathrooms', 'bedrooms', 'beds',
       'review_scores_rating', 'price_log', 'superhost', 'profile_pic'],
      dtype='object')
```

In [25]:
```
x = df2[['host_response_rate', 'host_acceptance_rate',
       'superhost', 'accommodates', 'bathrooms', 'bedrooms', 'beds',
       'review_scores_rating', 'profile_pic']]
y = df2['price_log']
```

In [26]:
```
lm1 = sm.OLS(y, x).fit()
lm1.summary()
```

Out[26]:

OLS Regression Results

Dep. Variable:	price_log	**R-squared:**	0.992
Model:	OLS	**Adj. R-squared:**	0.992
Method:	Least Squares	**F-statistic:**	5.304e+04
Date:	Tue, 23 Aug 2022	**Prob (F-statistic):**	0.00
Time:	08:58:41	**Log-Likelihood:**	1084.8
No. Observations:	3818	**AIC:**	-2152.
Df Residuals:	3809	**BIC:**	-2095.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
host_response_rate	0.0099	0.027	0.372	0.710	-0.042	0.062
host_acceptance_rate	1.3800	0.074	18.767	0.000	1.236	1.524
superhost	0.0136	0.008	1.804	0.071	-0.001	0.028
accommodates	0.0731	0.003	23.090	0.000	0.067	0.079
bathrooms	0.0251	0.006	3.916	0.000	0.013	0.038
bedrooms	0.0401	0.006	6.905	0.000	0.029	0.052
beds	-0.0161	0.005	-3.028	0.002	-0.027	-0.006
review_scores_rating	0.0031	0.000	6.415	0.000	0.002	0.004
profile_pic	0.0419	0.058	0.724	0.469	-0.072	0.155

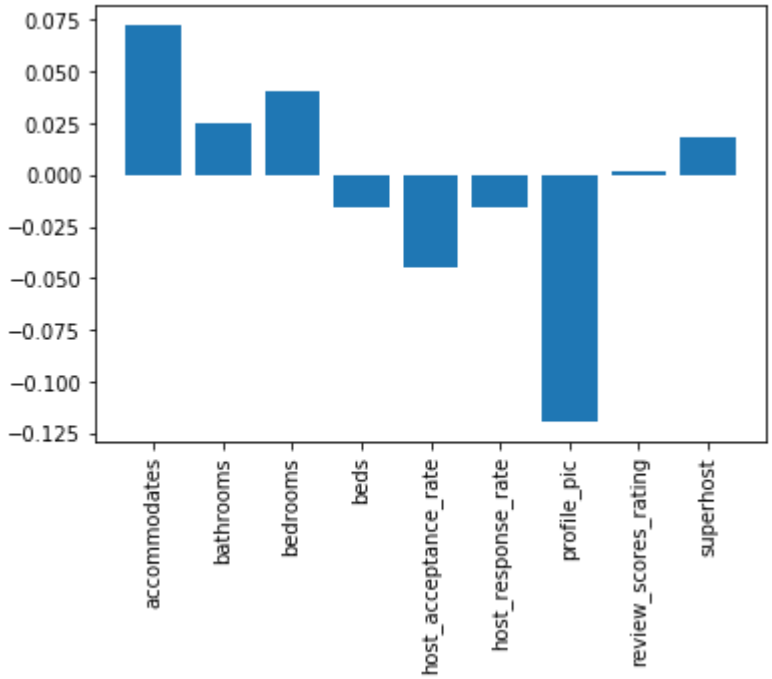
Omnibus:	161.839	**Durbin-Watson:**	1.608
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	439.267
Skew:	0.182	**Prob(JB):**	4.12e-96
Kurtosis:	4.621	**Cond. No.**	2.83e+03

The model result is not surpriced, in which most of factors have positive relationship with price, but the bed has the negative relationship with price.

Features of Importance

In [27]:

```
regr = linear_model.LinearRegression()
regr.fit(x, y)

importance = pd.DataFrame(data={
    'Attribute':x.columns,
    'Importance': regr.coef_
})
importance2 = importance.sort_values('Importance', ascending = False)

plt.bar(importance2['Attribute'], importance2['Importance'])
plt.xticks(rotation = 'vertical')
plt.show()
```



In [28]:
```
importance2.sort_values('Importance', ascending = False)
```

Out[28]:

	Attribute	Importance
3	accommodates	0.072470
5	bedrooms	0.040739
4	bathrooms	0.024576
2	superhost	0.018123
7	review_scores_rating	0.002089
0	host_response_rate	-0.015323
6	beds	-0.015590
1	host_acceptance_rate	-0.044528
8	profile_pic	-0.119139

The most important features are bathroom number, bedroom number, accommodates.

- Re-run regression with important features

In [29]:
```
x2 = df2[['accommodates', 'bathrooms', 'bedrooms',
       'superhost']]
y = df2['price_log']
```

In [30]:
```
lm1 = sm.OLS(y, x2).fit()
lm1.summary()
```

Out[30]:

OLS Regression Results

Dep. Variable:	price_log	**R-squared:**	0.879
Model:	OLS	**Adj. R-squared:**	0.879
Method:	Least Squares	**F-statistic:**	6943.
Date:	Tue, 23 Aug 2022	**Prob (F-statistic):**	0.00
Time:	08:58:41	**Log-Likelihood:**	-4116.7
No. Observations:	3818	**AIC:**	8241.
Df Residuals:	3814	**BIC:**	8266.
Df Model:	4		
Covariance Type:	nonrobust		

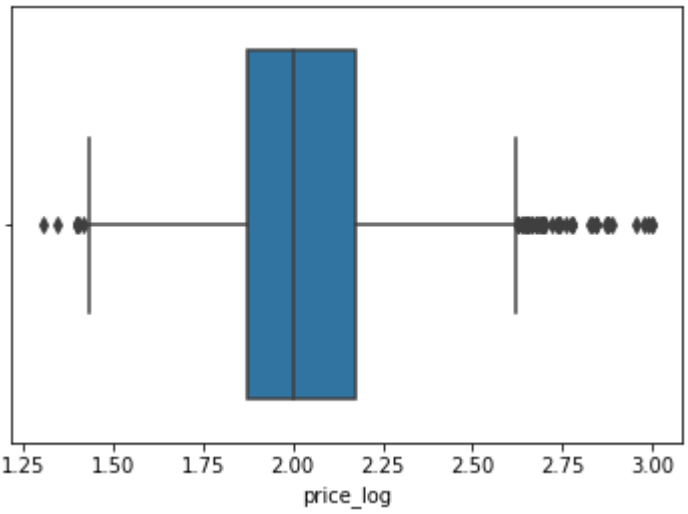
	coef	std err	t	P>|t|	[0.025	0.975]
accommodates	0.2087	0.009	23.430	0.000	0.191	0.226
bathrooms	0.8801	0.020	43.198	0.000	0.840	0.920
bedrooms	-0.0941	0.022	-4.297	0.000	-0.137	-0.051
superhost	0.3661	0.028	13.056	0.000	0.311	0.421

Omnibus:	2065.654	**Durbin-Watson:**	1.649
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	21038.151
Skew:	-2.385	**Prob(JB):**	0.00
Kurtosis:	13.464	**Cond. No.**	10.9

Words vs. Price

In [31]:
```
sns.boxplot('price_log', data=df2)
```

Out[31]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x201dbae86a0>
```



In [32]:
```
# calendar.head(1)
# listings2
reviews.head(1)
```

Out[32]:

	listing_id	review_id	date	reviewer_id	reviewer_name	comments
0	7202016	38917982	2015-07-19	28943674	Bianca	Cute and cozy place. Perfect location to every...

In [33]:
```
import nltk
nltk.download('vader_lexicon')
nltk.download('punkt')

from nltk import word_tokenize
from nltk.sentiment import SentimentIntensityAnalyzer
import operator
sia = SentimentIntensityAnalyzer()

from nltk.tokenize import TweetTokenizer
tweet = TweetTokenizer()

from nltk.corpus import stopwords
nltk.download('stopwords')
S = set(stopwords.words('english'))

from nltk.stem import WordNetLemmatizer
nltk.download('wordnet')
lemma = WordNetLemmatizer()

from nltk.stem import PorterStemmer
nltk.download('PorterStemmer')
ps = PorterStemmer()
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\uswangze\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\uswangze\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\uswangze\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\uswangze\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Error loading PorterStemmer: Package 'PorterStemmer' not
[nltk_data]     found in index
```

```
C:\Anaconda3\lib\site-packages\nltk\twitter\__init__.py:20: UserWarning: The tw
ython library has not been installed. Some functionality from the twitter packa
ge will not be available.
  warnings.warn("The twython library has not been installed. "
```

In [34]:
```
reviews_2 = reviews

reviews_2['comments'] = np.where(reviews_2['comments'].isna(), " ", reviews_2['co
```

In [35]:
```
reviews_2.comments.isna().sum()
```

Out[35]:
```
0
```

In [36]:
```
# sample = reviews_2[:100]
# sample.head(1)

PreprocessedTest = []
for line in reviews_2['comments'].values:
    line = line.strip().lower()
    tokens = word_tokenize(line)
    tokens_out = []
    for token in tokens:
        if not token in S:
            token = lemma.lemmatize(token)
            token = ps.stem(token)
            tokens_out.append(token)
    line_out = ' '.join(tokens_out)
    PreprocessedTest.append(line_out)
reviews_2['PreprocessedTest'] = PreprocessedTest
reviews_2.head()
```

Out[36]:

	listing_id	review_id	date	reviewer_id	reviewer_name	comments	PreprocessedTest
0	7202016	38917982	2015-07-19	28943674	Bianca	Cute and cozy place. Perfect location to every...	cute cozi place . perfect locat everyth !
1	7202016	39087409	2015-07-20	32440555	Frank	Kelly has a great room in a very central locat...	kelli great room central locat . beauti build ...
2	7202016	39820030	2015-07-26	37722850	Ian	Very spacious apartment, and in a great neighb...	spaciou apart , great neighborhood . kind apar...
3	7202016	40813543	2015-08-02	33671805	George	Close to Seattle Center and all it has to offe...	close seattl center offer - ballet , theater ,...
4	7202016	41986501	2015-08-10	34959538	Ming	Kelly was a great host and very accommodating ...	kelli great host accommod great neighborhood ....

In [38]:
```
reviews_2['words_count'] = reviews_2['PreprocessedTest'].str.split(" ").str.len()
reviews_2[['Year', 'Month', 'Date']] = reviews_2['date'].str.split("-", expand =

year_list = ['2015', '2016']
reviews_3 = reviews_2[reviews_2.Year.isin(year_list)]
```

In [39]:
```
reviews_3['listing_id'] = reviews_3['listing_id'].astype(int)
df2['id'] = df2['id'].astype(int)
```

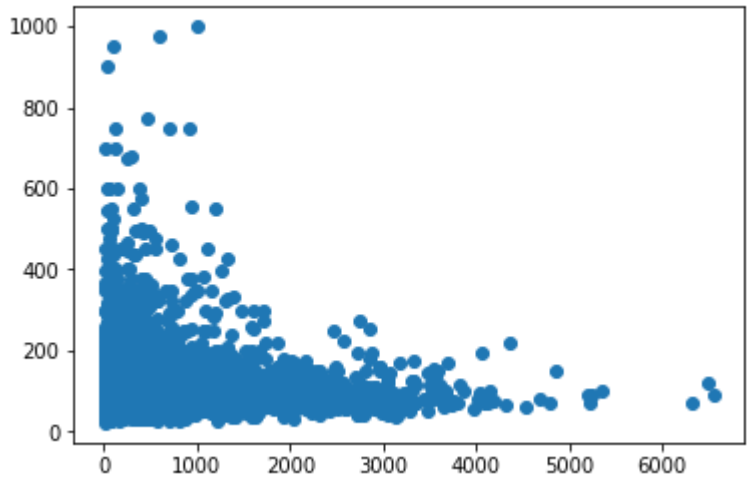
```
C:\Anaconda3\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
  """Entry point for launching an IPython kernel.
C:\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: 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#indexing-view-versus-copy (http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy)
```

In [40]:
```
reviews_4 = pd.DataFrame(reviews_3.groupby(['listing_id'])['words_count'].sum()).
reviews_4['listing_id'] = reviews_4['listing_id'].astype(int)
```

In [41]:
```
df_combine = pd.merge(df2, reviews_4, left_on = 'id', right_on = 'listing_id', ho
```

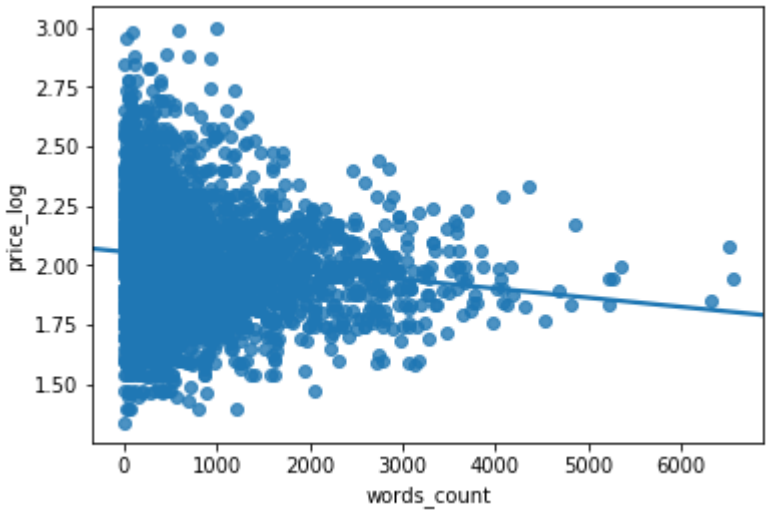
In [42]:
```
plt.scatter(df_combine['words_count'], df_combine['price'])
# plt.plot(df_combine['words_count'], df_combine['price'])
```

Out[42]:
```
<matplotlib.collections.PathCollection at 0x201e4c48e80>
```



In [43]:
```
sns.regplot(df_combine['words_count'], df_combine['price_log'], ci=None)
```

Out[43]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x201e4c90320>
```



In [44]:
```
x3 = df_combine[['accommodates', 'bathrooms', 'bedrooms', 'beds',
       'review_scores_rating', 'words_count']]
y = df_combine['price_log']
```

In [45]:
```
lm2 = sm.OLS(y, x3).fit()
lm2.summary()
```

Out[45]:

OLS Regression Results

Dep. Variable:	price_log	**R-squared:**	0.990
Model:	OLS	**Adj. R-squared:**	0.990
Method:	Least Squares	**F-statistic:**	5.145e+04
Date:	Tue, 23 Aug 2022	**Prob (F-statistic):**	0.00
Time:	09:04:37	**Log-Likelihood:**	531.85
No. Observations:	3111	**AIC:**	-1052.
Df Residuals:	3105	**BIC:**	-1015.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
accommodates	0.0836	0.004	21.569	0.000	0.076	0.091
bathrooms	0.0315	0.008	4.032	0.000	0.016	0.047
bedrooms	0.0311	0.007	4.280	0.000	0.017	0.045
beds	-0.0221	0.006	-3.418	0.001	-0.035	-0.009
review_scores_rating	0.0181	0.000	173.091	0.000	0.018	0.018
words_count	-1.402e-05	4.41e-06	-3.176	0.002	-2.27e-05	-5.36e-06

Omnibus:	333.856	**Durbin-Watson:**	1.697
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1122.091
Skew:	0.527	**Prob(JB):**	2.19e-244
Kurtosis:	5.747	**Cond. No.**	2.68e+03

In [ ]:
```
```