ruchigupta19 / **Boston-Airbnb-data-analysis**

No description, website, or topics provided.

1 commit **1** branch **0** releases **0** contributors

Branch: **master** New pull request Find file Clone or download

Bhavin Agarwal and **Bhavin Agarwal** boston airbnb Latest commit `a85fc49` on May 12, 2017

Analysis	boston airbnb	2 years ago
data	boston airbnb	2 years ago
extra	boston airbnb	2 years ago
readme.md	boston airbnb	2 years ago

readme.md

BOSTON AIRBNB OPEN DATA ANALYSIS

In this Project, I am going to explore the Airbnb data of Boston which includes 3 files:

- listings.csv
- calendar.csv
- reviews.csv

listings csv consists of details of all the listings in Boston including their price, accomodates, ratings, number of reviews, summary, name, owner name, Description, host Id and many other columns decribing details of listings.

calendar.csv consists of details of listings and its availability and its price.

reviews.csv consits of reveiws for each listing in Boston

This includes 5 analysis on Boston Airbnb open dataset with the help of which I tried to answer questions like What are the busiest times to visit Boston? busiest neighborhoods in Boston? seasonal pattern of airbnb housing price? and many more

ANALYSIS - 1

WHAT CAUSES DIFFERENCE IN PRICES OF LISTINGS?

Data Wrangling: I have collected, cleaned and transformed **listings.csv** data by levaraging pandas and numpy to be used for performing meaningful analysis.

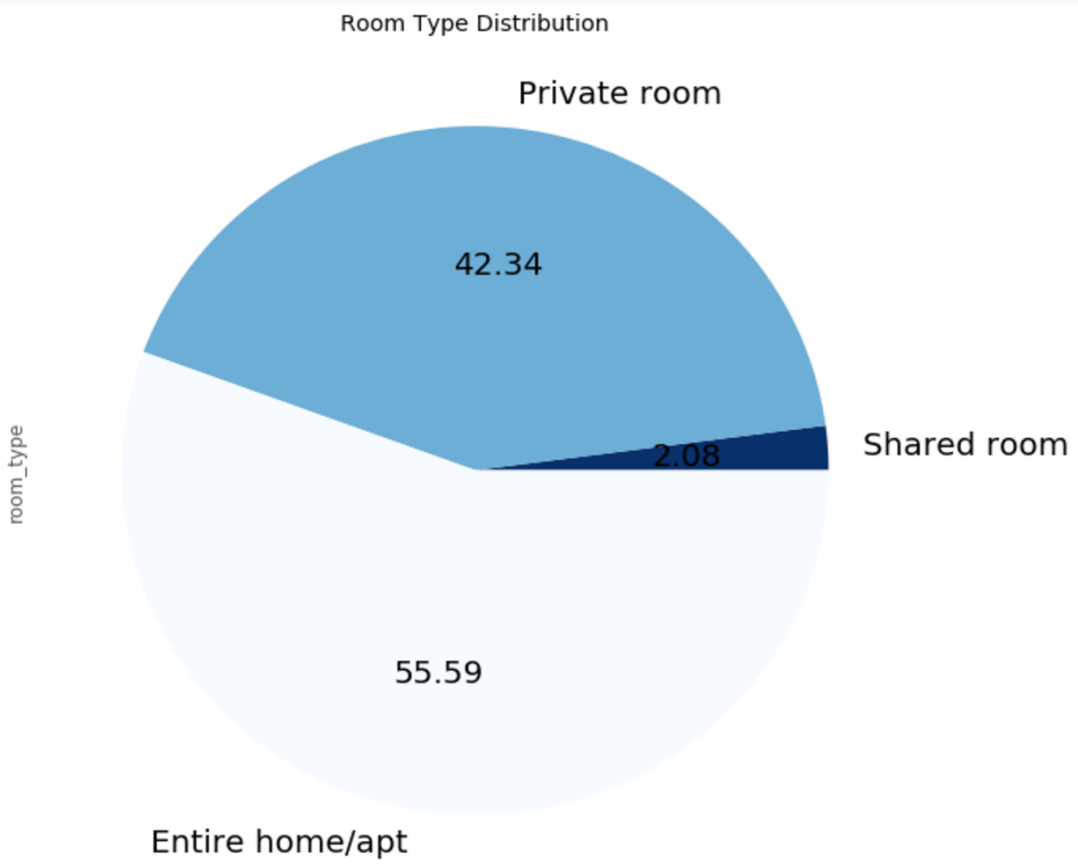
```
# replacing NaN values with 0
inputDF.fillna(0, inplace=True)

#clean the data to make it float
for p in price:
    p=float(p[1:].replace(',',''))
    prices.append(p)

#exclude the listings with 0 for price,beds,bedrooms,accomodates etc
inputDF = inputDF[inputDF.bathrooms >0]
inputDF = inputDF[inputDF.bedrooms > 0]
inputDF = inputDF[inputDF.beds > 0]
inputDF = inputDF[inputDF.price  > 0]
inputDF = inputDF[inputDF.review_scores_rating  > 0]
inputDF = inputDF[inputDF.reviews_per_month > 0]
inputDF = inputDF[inputDF.accommodates  > 0]
inputDF.head()
```

Cleaned and transformed data can be accessed here

then I categorized different listings based upon their room type which gave the following results:

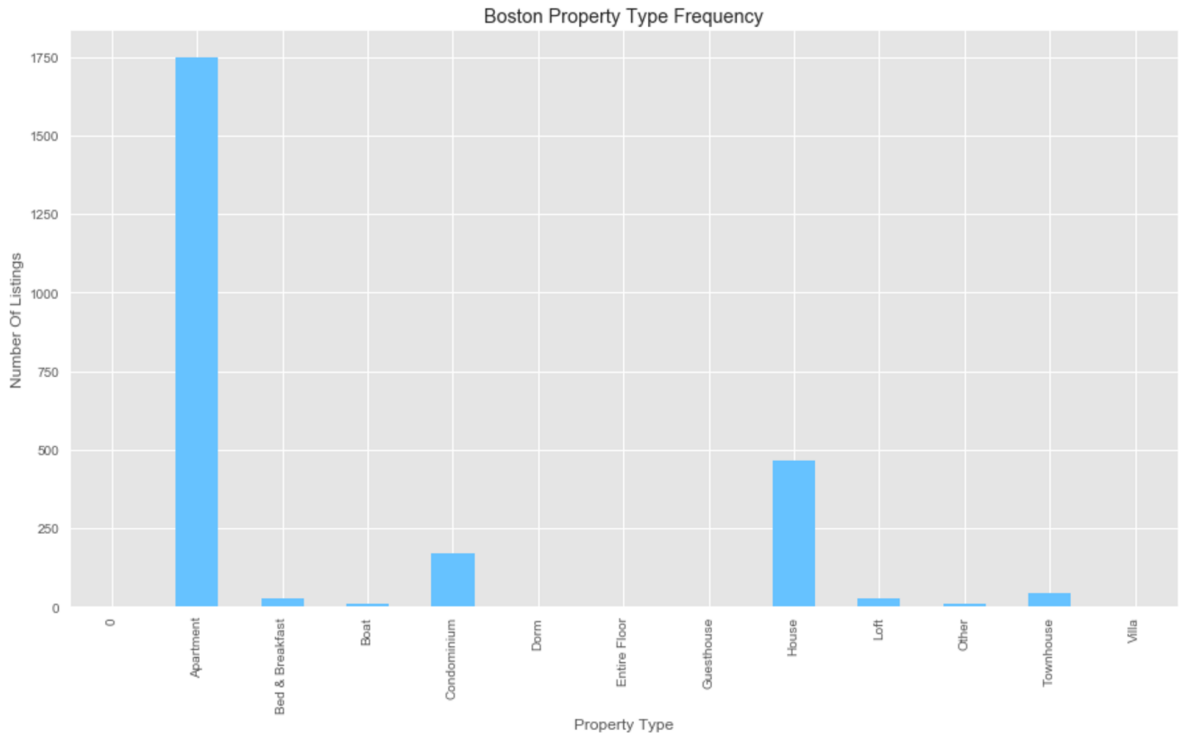
```
roomType_DF=inputDF.groupby('room_type').id.count()
```

Room Type	Number Of Listings
Entire home/apt	1393
Private room	1061
Shared room	52



I categorized different listings based upon their property type which gave the following results:

```
propertytype_DF = inputDF.groupby('property_type').id.count()
```



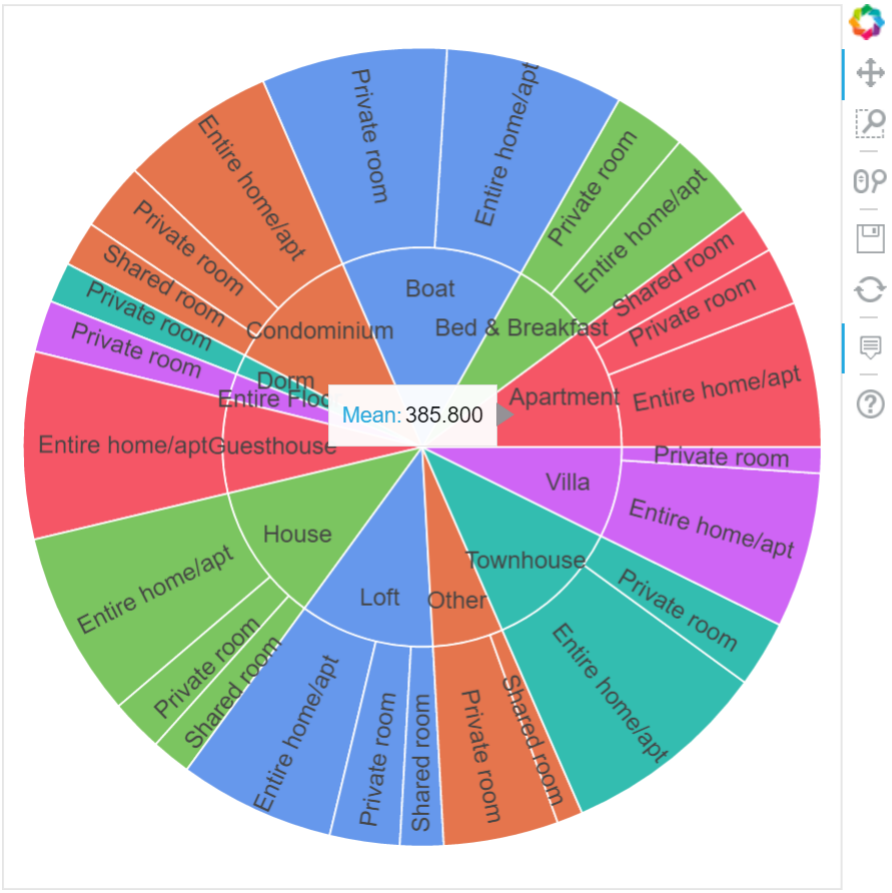
1st Data point:

It can be concluded that people are more inclined towards listing their entire property than that of private rooms or shared rooms.It can also be seen that property type also plays an important factor. Not surprisingly, Apartment and houses take up an overwhelming majority of all listings, although we do see few instances unfamiliar residencies here and there.

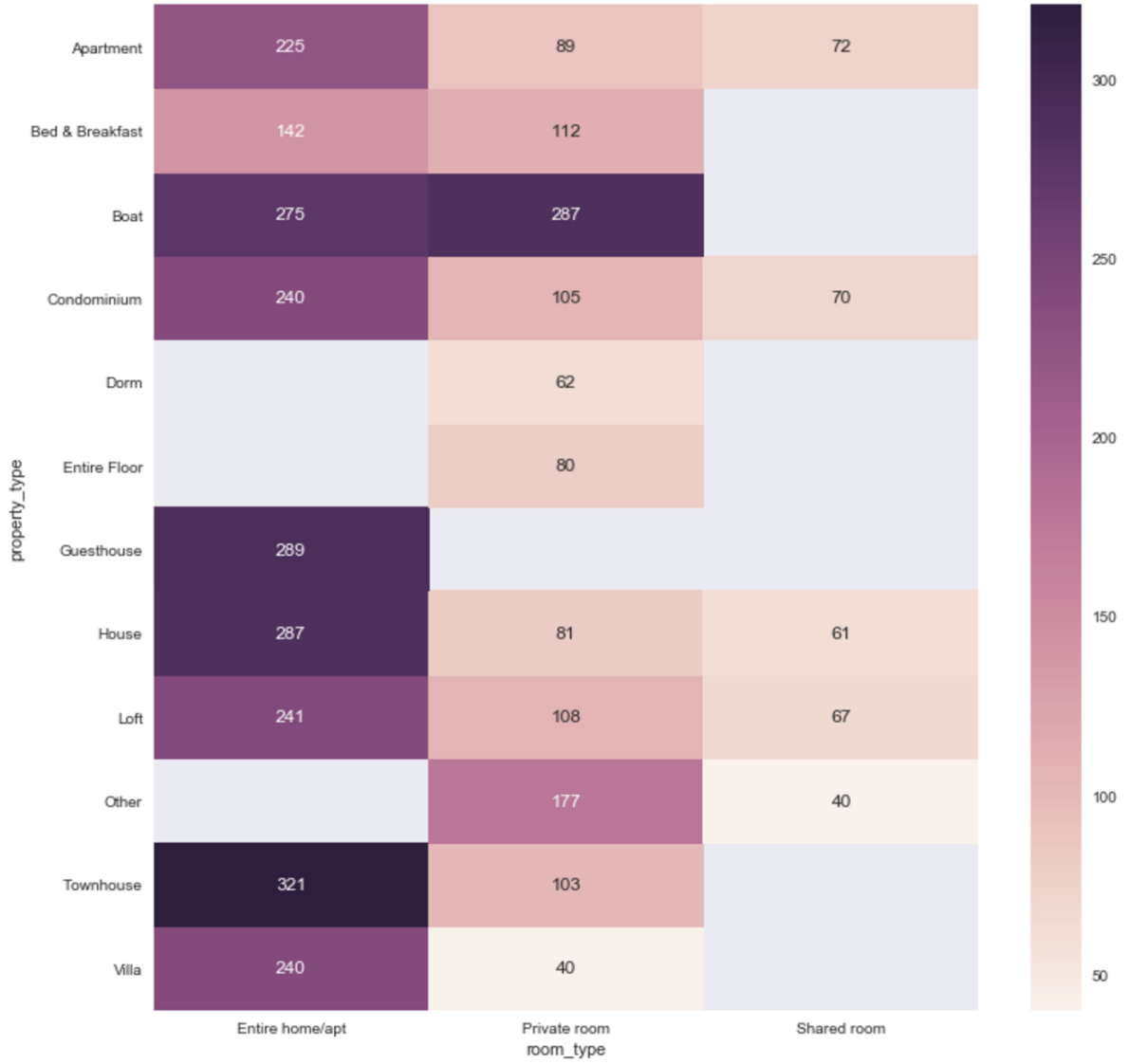
Now analyzing the prices for different room type and property type and plotted a donut chart for the same which includes category and subcategory for all those categories and hovering on each subcategory will show mean price for each of them

```
roomProperty_DF = inputDF.groupby(['property_type','room_type']).price.mean()
```

BokehJS 0.12.5 successfully loaded.



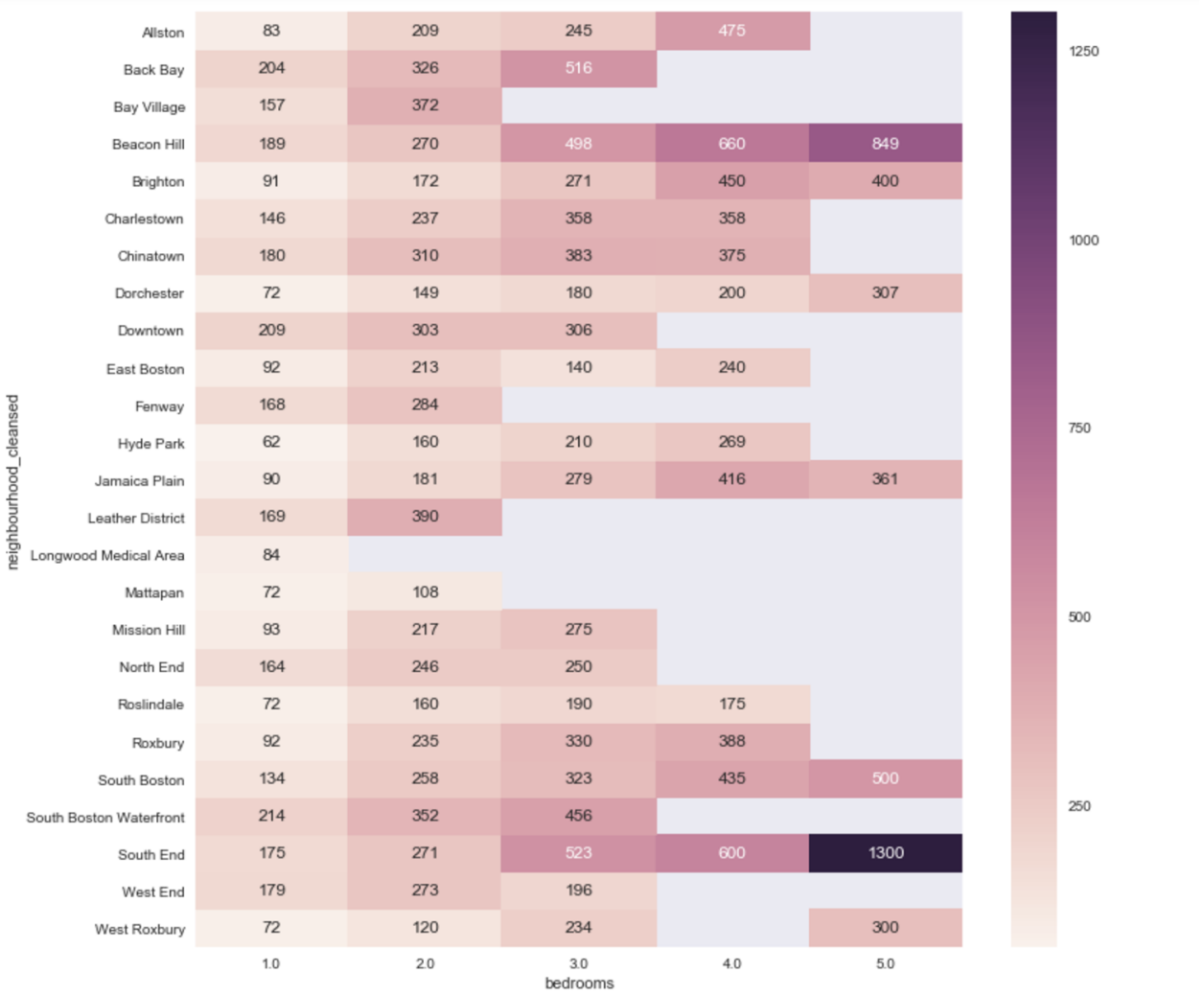
While plotting the same on a heat map:



2nd Data point:

This chart allows us to see all the listings' prices broken down by property type and room type. This gives us a much better understanding of the price breakdown in Boston based on property and room types. It can be analyzed that for almost all property type,prices for Entire home/apartment is the maximum.This tells us that Property type and room type plays a very important role in deciding price of a listing.

Lets see how the number of bedrooms available affects the price of a listing.



3rd Data Point:

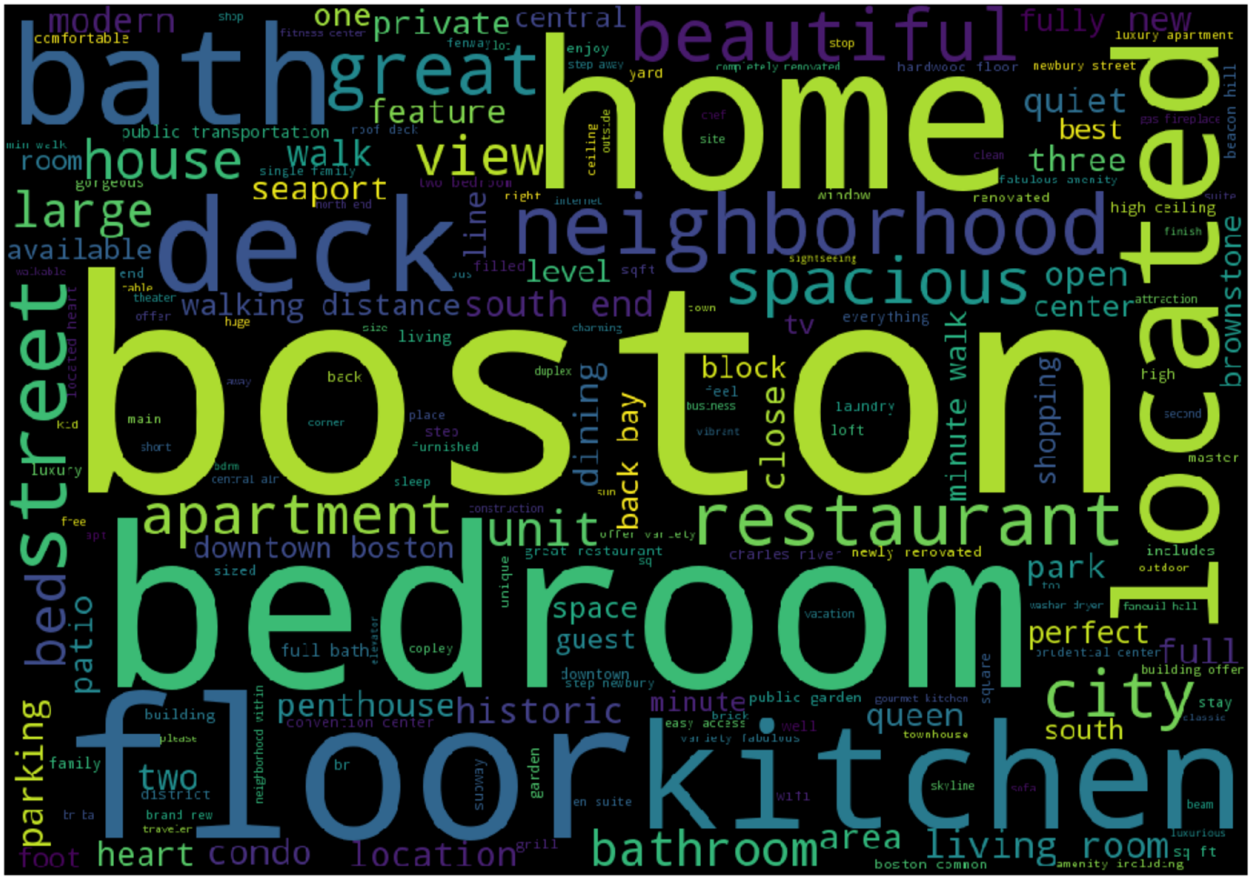
It can be analyzed that with the increase in the number of bedrooms price of listing increases.Although, it depends upon the neighbourhood as well.

So the analysis gives us data points that the prices of listings on Airbnb depends upon the room type, property type, number of bedrooms and neighbourhood.It can be seen that the property with type as Apartment and the listing as with type as entire house with maximum number of bedooms has highest price.Although it depends upon neighbourhood as well which is analyzed in the next analysis

Now that we've seen how property types and room types along with neighborhood can affect the listing prices, let's investigate the summary.

```
summaryDF = inputDF[['summary','price']]
summaryDF = summaryDF[pd.notnull(summaryDF['summary'])]
summaryDF = summaryDF[summaryDF['summary']!=0]
summaryDF = summaryDF.sort_values('price',ascending=[0])
top100DF = summaryDF.head(100)
top100DF.head()
```

This will give us summary and price for all the listings. I have chosen top 100 listings on the basis of price to find out what all common words are used by hosts while posting a listing on airbnb. I have created a wordcloud for the same and to plot it I have cleaned the data by removing punctuation, unwanted characters and numbers.After that I have also removed all the stopwords.I have lemmatized it using WordNetLemmatizer from NTLK and plotted a wordcloud for the same to visualize the most common words these hosts utilized to describe their listings on AIRBNB.

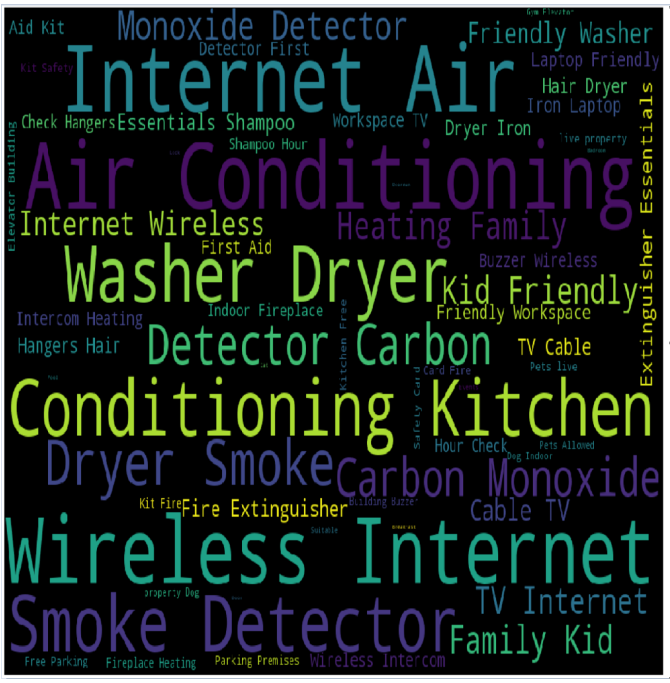


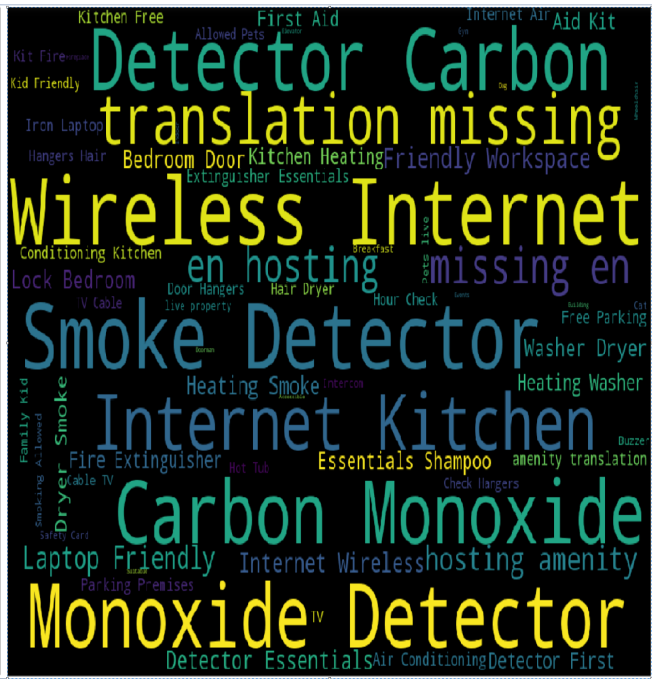
4th Data Point:

It can be seen that unique words like home, south Boston, bedroom, floor, kitchen, restaurant, spacious, neighborhood ,located are very commonly words used when hosts are describing their homes. This is making sense because the primary purpose of airbnb is not to provide luxury hotel suites but just a convinient place to stay.Naturally hosts understand these purposes and create their summaries based on location and requirements in order to attract as many travellers as they can. So if hosts are not able to attract too many travellers then they can add these keywords to their summaries in order to attract travellers to choose their listings.

Now trying to analyze how the amenities provided by the listing is related to the price of the same.To analyze the same I have plotted two diferent wordclouds which in turn helped me to find out what extra emenities are provided by listings with higher price.

```
#Data collection
for index,row in amenitiesDFtop.iterrows():
    p = re.sub('[^a-zA-Z]+',' ', row['amenities'])
    allemenities+=p

#tokenizing
allemenities_data=nltk.word_tokenize(allemenities)

#lemitizing
wnl = nltk.WordNetLemmatizer()
allemenities_data=[wnl.lemmatize(data) for data in filtered_data]
```





5th Data Point:

It can be clearly seen that listings with higher prices have extra eminities such as Air conditioning, washer/dryer, Kid friendly, Heating, hair dryer, buzzer and *Extra emenities comes with extra prices*

Conclusion:

It can be concluded that prices of listings depends upon following factors:

1. The type of room chosen by the traveller and mostly booking an Entire property costs maximum followed by private room and shared apartment.
2. The type of property chosen by the traveller and it can be analyzed that Townhouse and houses are the properties with maximum prices and apartments, houses take up an overwhelming majority of all listings.
3. Price of a listing also depends upon the number of bedrooms the property have and the same also depends upon the neighborhood of the property
4. the summary section is the one which helps to attract travellers and analyzed that presence of unique words like home, south Boston, bedroom, floor, kitchen, restaurant, spacious, neighborhood ,located words tends to attract more travellers
5. with the increase in prices the eminities provided by host also increases.

ANALYSIS - 2

WHERE TO INVEST A PROPERTY IN BOSTON TO GET MAXIMUM RETURNS FROM AIRBNB?

It has been analyzed earlier that the maximum number of listings are for Entire Home/Apartment. Lets check average prices for these listings based on room type.

```
# Average prices for each type of listing

avgPrice_DF=inputDF.groupby('room_type').price.mean()
avgPrice_DF=avgPrice_DF.reset_index()
avgPrice_DF=avgPrice_DF.rename(columns={'price':'average_Price'})
avgPrice_DF
```

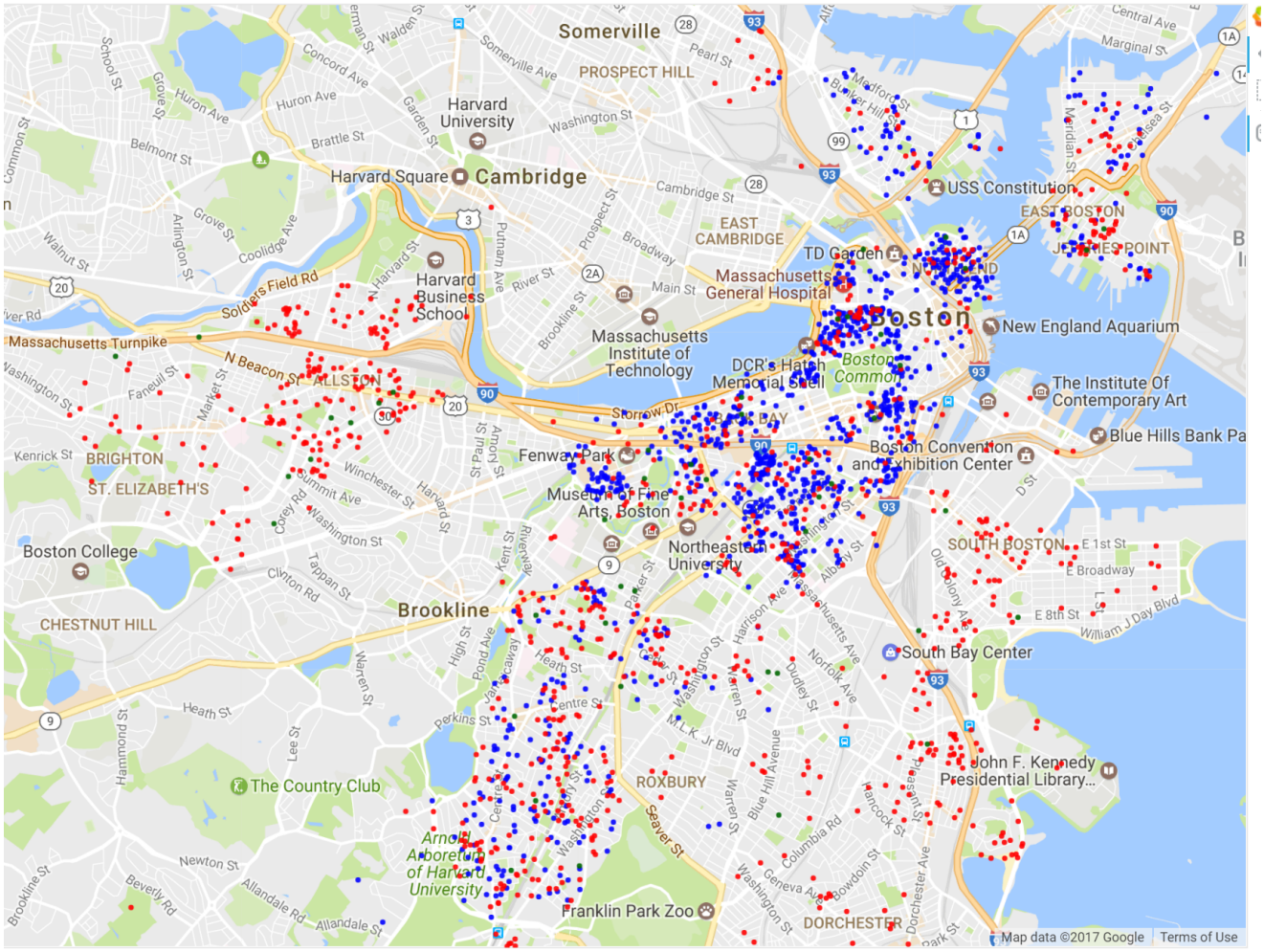
Room Type	Average Price
Entire home/apt	232.322326
Private room	89.505184
Shared room	69.903846

1st Data Point:

It clearly shows that Average Price of Entire home/Apartment is very high when compared to Private room and shared room.It can be concluded that Entire home/Apartment room type has the maximum average price which gives us a data point that after buying an Apartment, listing it as an entire Apartment on Airbnb will help to generate maximum revenue.

I have plotted Geographical Clusters to find out which area in Boston has maximum listings on Airbnb and to get a better understand of which is best neighborhood to invest a property.

```
# seggregating each type of property

home = inputDF[(inputDF.room_type == 'Entire home/apt')]
private = inputDF[(inputDF.room_type == 'Private room')]
shared = inputDF[(inputDF.room_type == 'Shared room')]

location_home = home[['latitude', 'longitude']]
location_private = private[['latitude', 'longitude']]
location_shared = shared[['latitude', 'longitude']]
```



where,

1. Blue circles indicates Entire Apartment/home listings

2. red circle indicates private room listings
3. green indicates shared listings

It can be seen that Wealthier neighborhoods, mainly South End, north end, Fenway park and back bay have a higher proportion of listings for an entire apartment while Less central and poorer neighborhoods have a disproportionate number of listings from multi-unit listers.For example, Allston, there are only listings with private room.

2nd Data Point:

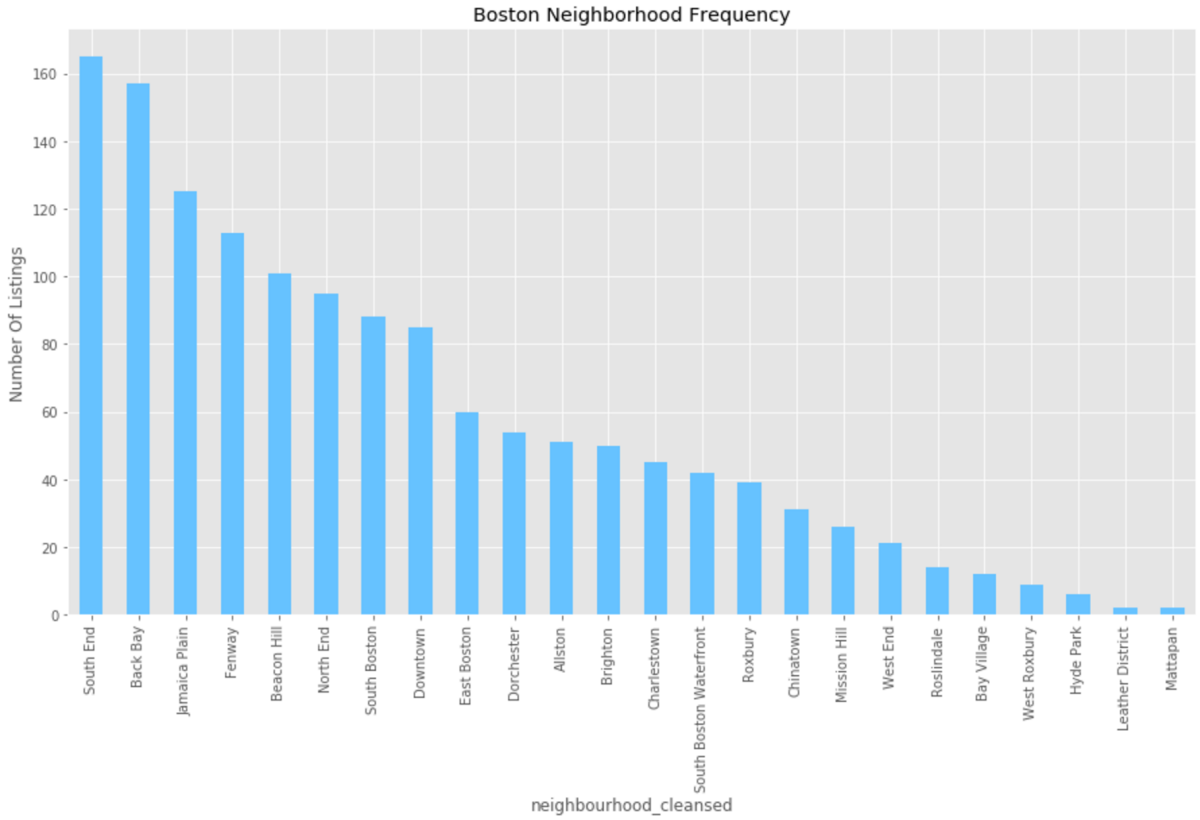
this analysis gives us another data point that Wealthier neighborhoods should be chosen for property investment.

Analyzing data on Entire Apartments/home to find the right location to invest property.

```
# grouping neighbourhood by number of listings

neighbourhood_DF=home.groupby('neighbourhood_cleansed').id.count()


# grouping neighbourhood by average price of listings

neighbourhoodPrice_DF=home.groupby('neighbourhood_cleansed').price.mean()


#Merging above two dataframes

mergeDF=pd.merge(neighbourhood_DF,neighbourhoodPrice_DF,on='neighbourhood_cleansed')
mergeDF.head()
```

neighbourhood_cleansed	Number_Of_Listings	Average_Price
South End	165	245.290909
Back Bay	157	280.006369
Jamaica Plain	125	211.160000
Fenway	113	240.398230
Beacon Hill	101	255.178218

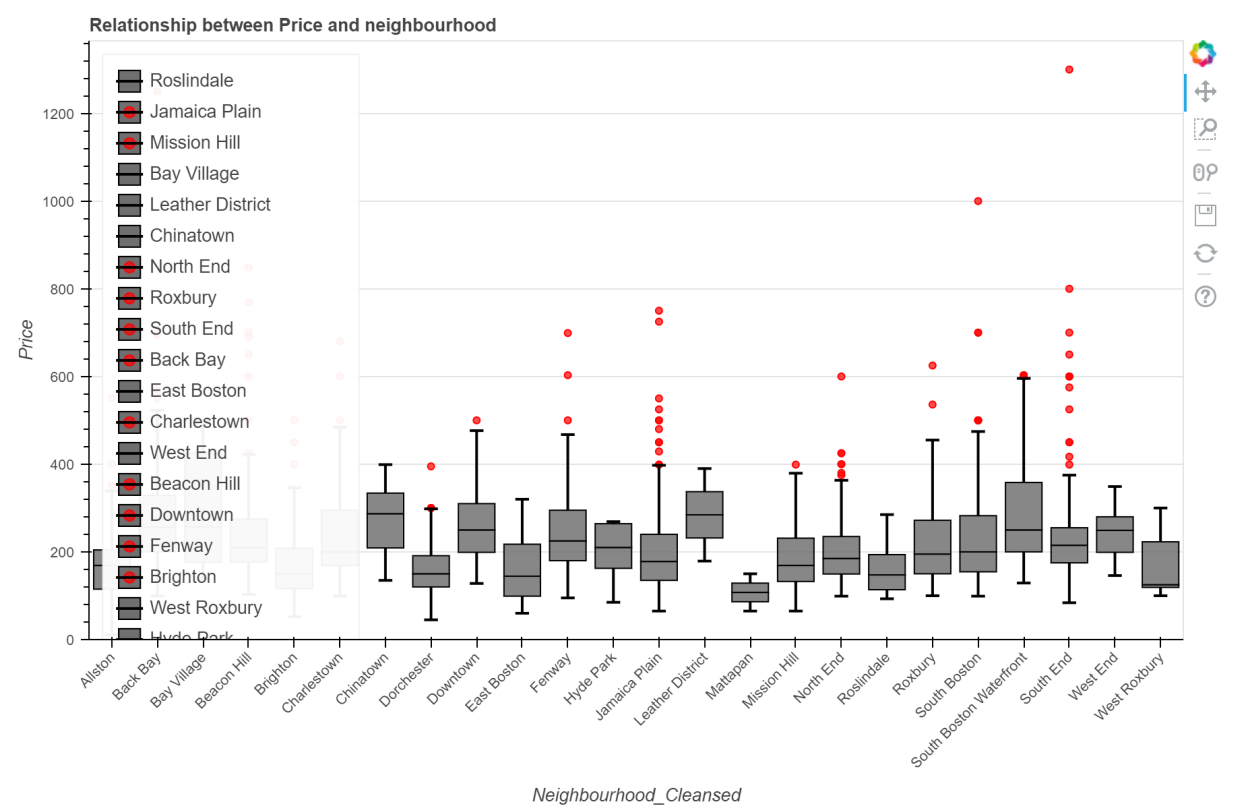
visualizing the frequency of listings on the basis of neighbourhood where room type is entire apartment



3rd Data Point:

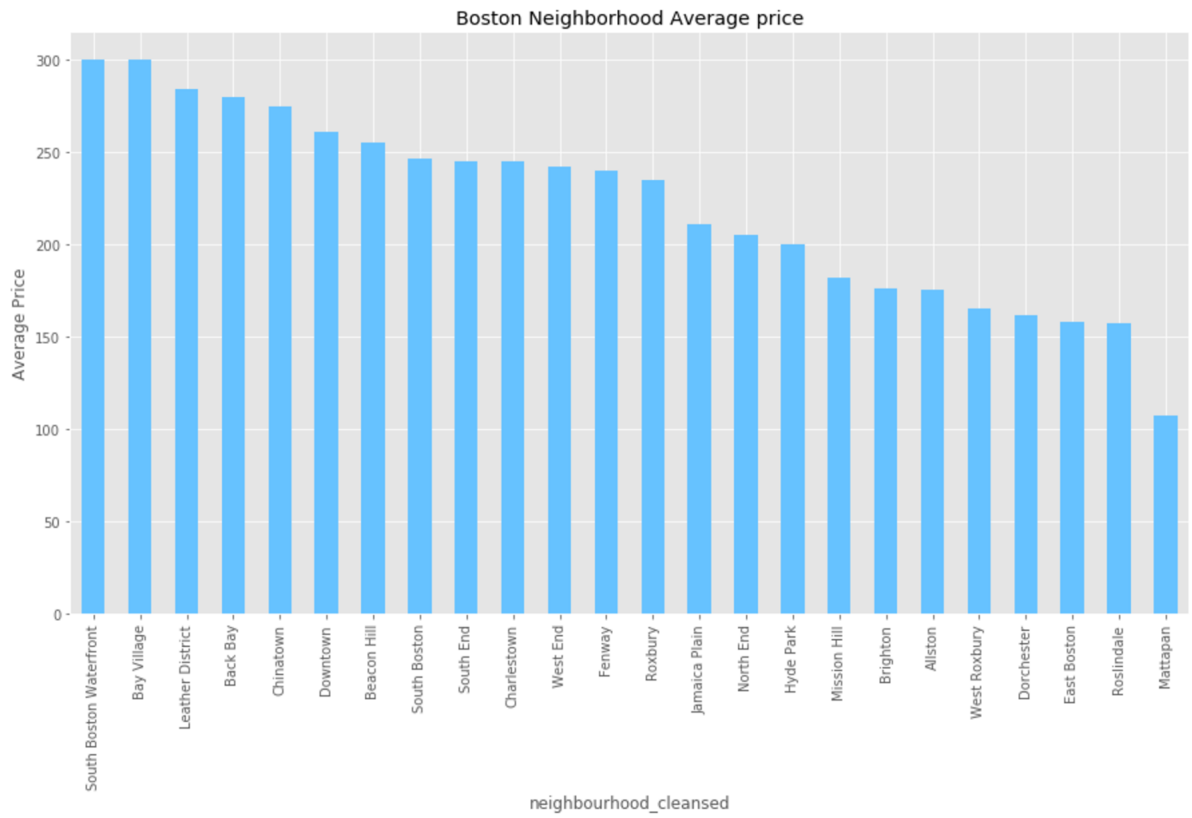
It can be noticed that the maximum number of listings for entire apartment for Boston on Airbnb is in South End area followed by back bay and Jamaica plain.So these area become potential neighbourhood to invest a property

Exploring the relationship between price and neighbourhood



It can be analyzed that the maximum price is reached for Back Bay and soutn end.

Now lets visualize average price of these listings on the basis of neighbourhood where room type is entire apartment



4th Data Point:

By analyzing the number of listings and prices per neighborhood, we can get a clearer understanding of how accurate the average price is per neighborhood. The neighorhoods with a lot of listings (such as South End & Back Bay), we can expect a more accurate average prices. However, neighborhoods with less than 100 listings might have relatively inaccurate average prices due to presence of outliers. Looking at the analysis done so far, one might conclude that certain neighborhoods are more 'expensive' than others.

For next analysis I have compiled a list of keywords for differnt neighborhoods in Boston which should be included while writing description about the listing on Airbnb.

```
#function to clean the data and compile a list of most common words
def cleanData(neighbrhood_name,descrip):
    p = re.sub('[^a-zA-Z]+',' ', descrip)
    cmn_words=['The','I','Boston','room']
    descrip_data=nltk.word_tokenize(p)
    filtered_data=[word.lower() for word in descrip_data if word not in cmn_words if word not in
stopwords.words('english')]
    wnl = nltk.WordNetLemmatizer()
    counts=Counter([wnl.lemmatize(data) for data in filtered_data])
    commn_words=[]
    for w in counts.most_common(5):
        commn_words.append(w[0])
    return ' '.join(commn_words)


#calling function and adding returned data to dataframe
for a in summ.items():
    top5words=cleanData(a[0],a[1])
    final_DF_neighbrhood=final_DF_neighbrhood.append(pd.Series([a[0],top5words],index=['neighborhood','top 5 words
in description']),ignore_index=True)
```

neighborhood	top 5 words in description
Roslindale	roslindale minute guest house neighborhood
Jamaica Plain	jamaica walk minute neighborhood bedroom
Mission Hill	apartment bed bedroom walk medical
Longwood Medical Area	longwood bathroom access kitchen medical
Bay Village	walk bay bed arlington bedroom

Click here to have a look at complete list

5th Data Point:

I can be observed that for neighborhood "South Boston Waterfront" major words include seaport, "Hyde Park" includes quiet, "East Boston" includes airport, downtown, station and chinatown includes theatre.All these observations shows that these words are unique to a particular neighborhood which gives us another data point that while listing the property on airbnb these words should definitely be included to attract travellers.

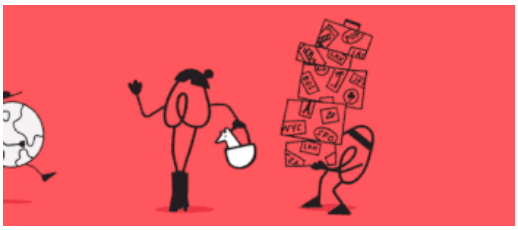
Conclusion:

After combining all the data points collected from above analysis it can be concluded that:

1. Invest a property in a wealthy neighborhood and according to this analysis that neighborhood should be 'Back Bay' and 'South End' as they have maximum number of listings and average prices in these neighborhood is fairly high.
2. After buying an Apartment, listing an entire Apartment on Airbnb will help to generate maximum revenue.
3. While listing the property on AirBnB, keywords describing the area should be added in order to gain the attention of travellers

It needs to be analyzed that how prices of listings vary with seasons which constitutes the next analysis in which I tried to find out how prices vary according to season,month,week,day

ANALYSIS - 3



SEASONAL PATTERN OF PRICES

Data Wrangling: I have collected, cleaned and transformed **calendar.csv** data by levaraging pandas and numpy to be used for performing meaningful analysis.

```
#replacing NaN values with 0
calendarDF.fillna(0, inplace=True)
calendarDF = calendarDF[calendarDF.price != 0]
```

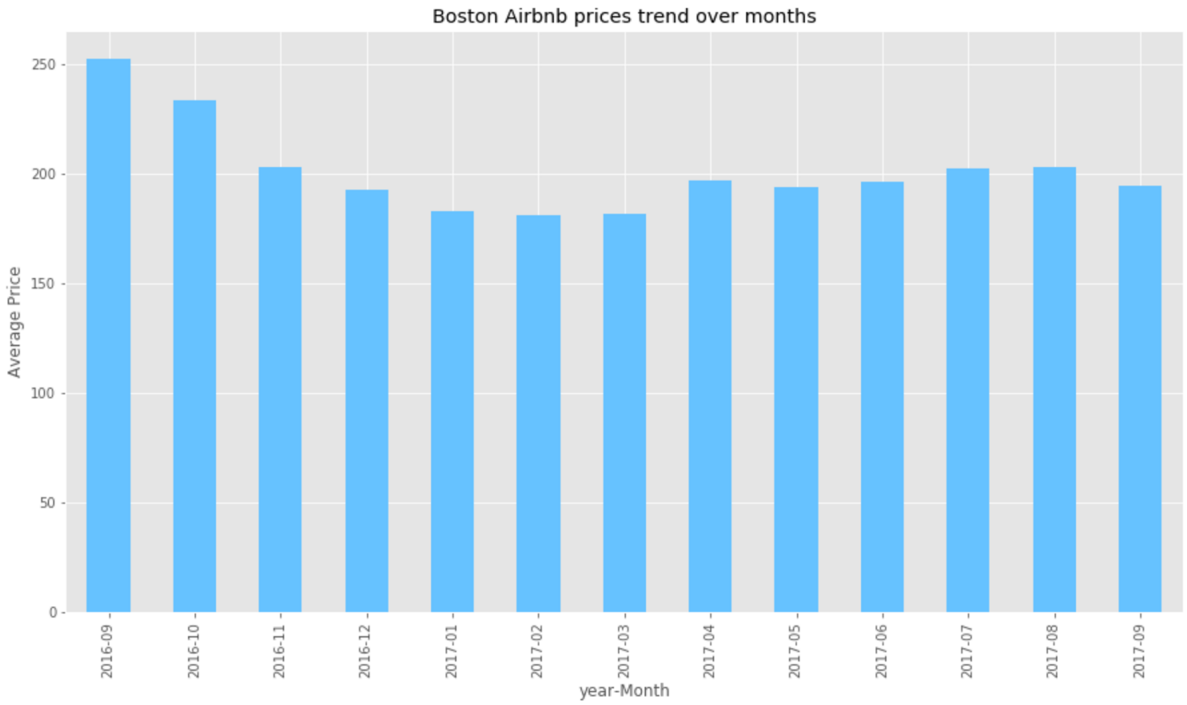
```
#Extracting prices from the table
price = calendarDF['price']
prices=[]

for p in price:
    p = re.sub('[^0-9.]+','', p)
    prices.append(float(p))
```

Cleaning of data has been done and added new columns namely Year,Month and Day by splitting date.I analyzed the data by group it on the basis of Year and Month to see the trend of prices.

```
yearDF=calendarDF.groupby(['Year','Month']).price.mean()
```

Data can be accessed from here

It can be seen that the data is available from September 2016 to September 2017 and when average prices are analyzed maximum rates for the listings were in the month of september.Visualizing the same for a better understanding.



1st Data Point:

It can be clearly seen that the maximum average price for listings were in the month of september and October 2016 and the reason is because of good weather and Massachussetts' best time to observe fall colors.Fall Colors in Massachusetts attracts a lot of visitors which makes September and October peak months for Airbnb hosts.

To analyze the data further, I have extracted name of the day from given date and checked weather it was a holiday and what is the reason for that holiday using datetime, clendar and holidays modules of python.

```
calendarDF.fillna(0, inplace=True)
us_holidays = holidays.US()


for index,row in calendarDF.iterrows():
    sdate = datetime.date(int(row['Year']),int(row['Month']),int(row['Day']))
    vall=date(int(row['Year']),int(row['Month']),int(row['Day'])) in us_holidays
    calendarDF.set_value(index,'day_Name',calendar.day_name[sdate.weekday()])
    calendarDF.set_value(index,'holiday',vall)
    calendarDF.set_value(index,'us_holidays_name',us_holidays.get(sdate))
```

Output data consisting days and holidays information can be accessed here

Added 3 new columns Day_Name, Holiday and us_holiday_name which counsits of name of the day, boolean value for "is it a holiday?" and reason for the holiday respectively.

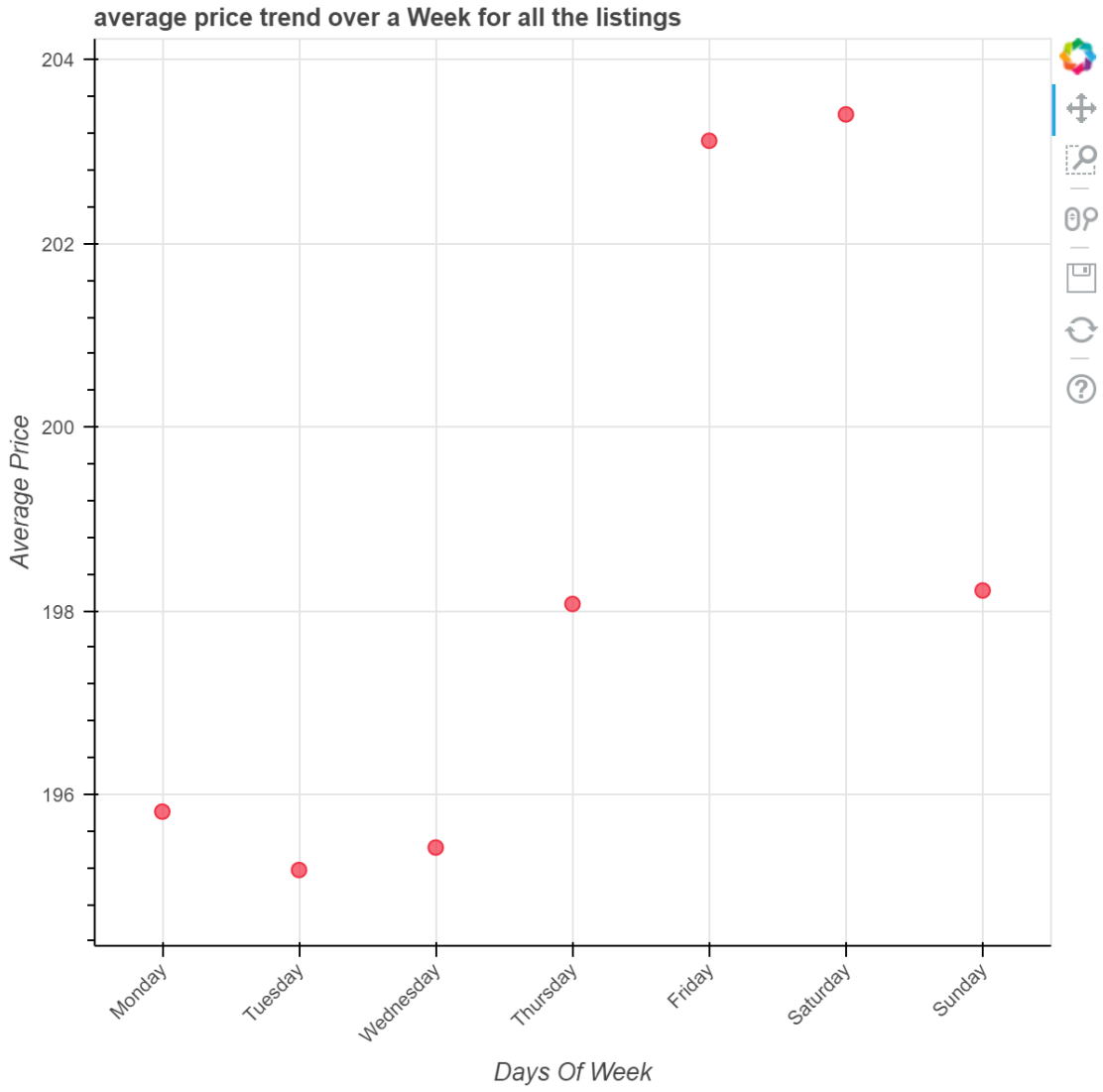
Trying to find a trend of avergae prices for a week

```
#calculating Average price for each day
```

```
dayDF=calendarDF.groupby('day_Name').price.mean()
```

day_Name	Average_Price	day_num
Monday	195.809561	1
Tuesday	195.173842	2
Wednesday	195.418228	3
Thursday	198.073112	4
Friday	203.121167	5
Saturday	203.408387	6
Sunday	198.219764	7

It can be seen that the average price of listings increases on weekends and are usual on weekdays.Lets plot it to get a better understanding

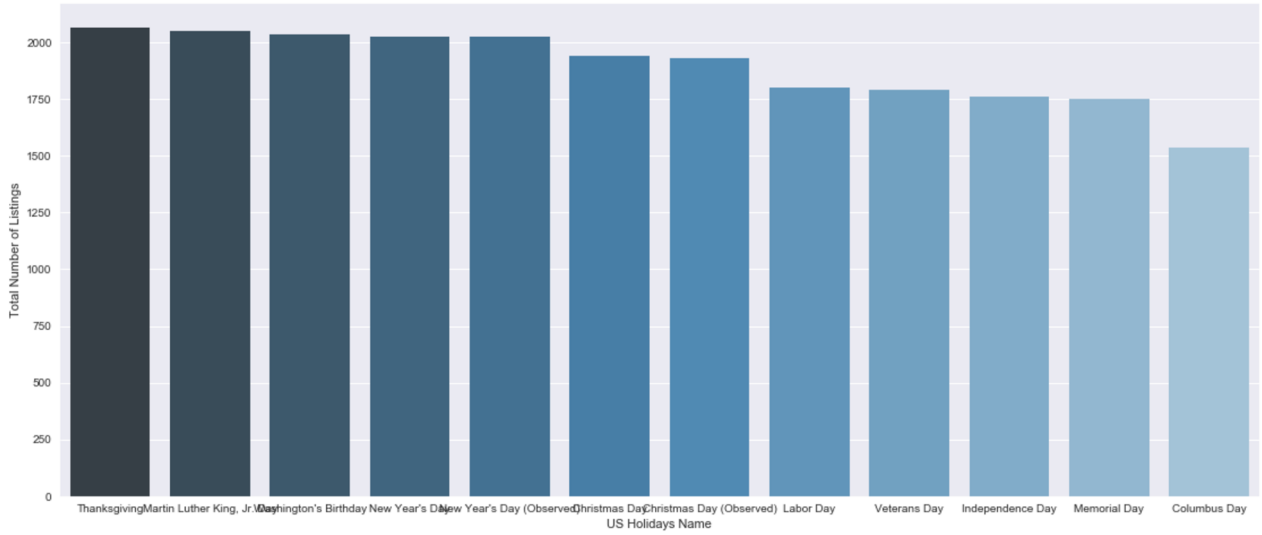


2nd Data Point

It can be seen that the prices are fairly high for the weekends than that of weekdays.Now we need to dig in into Sept 2016 and oct 2016 data to find the reason behind increase in Average prices.

Analyzing all the holidays

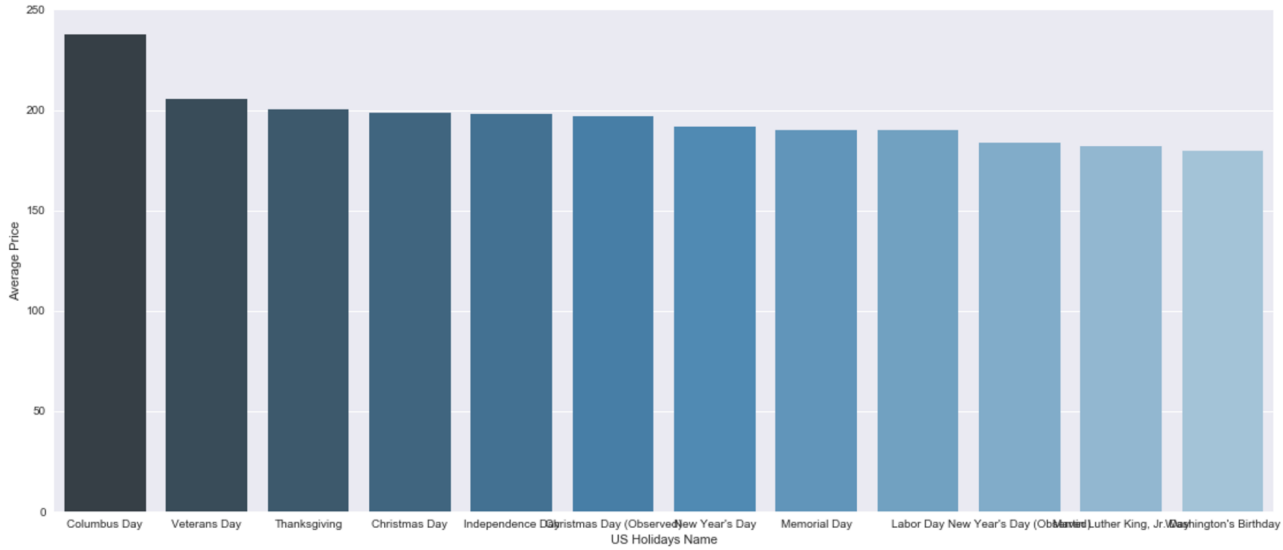
```
#checking which holiday has maximum listings

holidayDF=calendarDF.groupby('us_holidays_name').listing_id.count()
```



It can be seen that the maximum number of listings is for thanksgiving which can be reasoned as its a very popular holiday.Lets dig in further to find which holiday has the maximum average price.

```
holidayPriceDF=calendarDF.groupby('us_holidays_name').price.mean()
```

us_holidays_name	price
Columbus Day	237.838101
Veterans Day	205.283324
Thanksgiving	200.077407
Christmas Day	198.380805
Independence Day	197.900737
Christmas Day (Observed)	196.901139
New Year's Day	191.555994
Memorial Day	190.034305
Labor Day	189.907829
New Year's Day (Observed)	183.823384



3rd Data Point:

It can be observed that the maximum average price of listings is for **columbus day** and that mean Columbus day is the busiest US long weekend holiday for Boston.

Lets dig in into september 2016 and october 2016 data to find the reason for increase in average prices.

```
#analyzing data of september and october

checkHolidaysDF=calendarDF[(calendarDF['Year'] == '2016') & ((calendarDF['Month'] == '09') |(calendarDF['Month'] ==
'10'))]
checkHolidaysDF=checkHolidaysDF.groupby(['Year','Month','holiday']).price.mean()
```

Year	Month	holiday	price
2016	09	False	252.677308
2016	10	False	233.264468
2016	10	True	237.838101

```
#analyzing longweekednd holiday days

columbusDF=calendarDF[(calendarDF['Year'] == '2016') & (calendarDF['Month'] == '10' ) & ((calendarDF['Day'] ==
'08') | (calendarDF['Day'] == '09') | (calendarDF['Day'] == '10'))]
columbusDF.groupby('Month').price.mean()
```

Output: 244.633573

```
#analyzing rest of the days

NocolumbusDF=calendarDF[(calendarDF['Year'] == '2016') & (calendarDF['Month'] == '10' ) & ((calendarDF['Day'] !=
'08') | (calendarDF['Day'] != '09') | (calendarDF['Day'] != '10'))]
NocolumbusDF.groupby('Month').price.mean()
```

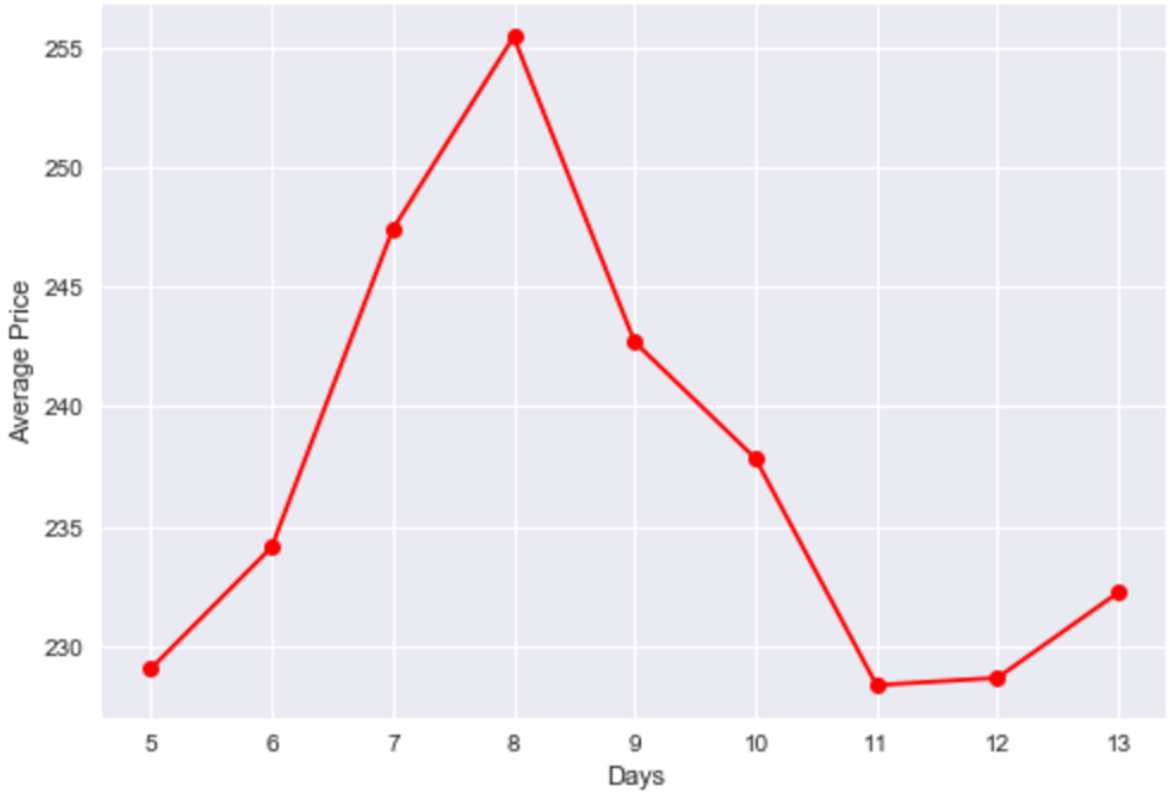
Output: 233.416248

4th Data Point:

It can be analyzed that on a long weekend the average price is 244.63 while for the whole month the average price was 233.Hence, this suggests that the price of listings increses with the presence of long weekend.

analyzing data from date 5th of October to date 13th of October which includes both long weekend and normal workdays

```
octDF=calendarDF[(calendarDF['Year'] == '2016') & (calendarDF['Month'] == '10' )& ((calendarDF['Day'] == '05' )|
(calendarDF['Day'] == '06' )| (calendarDF['Day'] == '07' )| (calendarDF['Day'] == '08' )| (calendarDF['Day'] ==
'09' )| (calendarDF['Day'] == '10' )| (calendarDF['Day'] == '10' )| (calendarDF['Day'] == '11' )|
(calendarDF['Day'] == '12' )| (calendarDF['Day'] == '13' ))]
octDF=octDF.groupby('Day').price.mean()
```



hence this graph clearly shows that Long weekends have higher prices of listings than that of normal days which gives us another data point that price of a listing increases with the presence of Long weekend.

Conclusion:

After adding up all the analyis done so far for observing seasonal changes in prices,it can be concluded that:

1. If a traveller is visiting Boston on a low budget then he should avoid visiting during September and October as these are the times when average price of listings are fairly high as compared to the other months.
2. If the taveller is on low budget then he should avoid visiting Boston on Weekends as prices of listings on weekend are higher than that of weekdays.
3. Prices of listings also depends upon US holidays and longweeknds.Prices on long weekends and US holidays are higher than compared to other days.So for a low budget travel, a traveller should avoid travelling on a long weekend,Specially the long weekend of October i.e columbus day.

ANALYSIS - 4

SENTIMENT ANALYSIS OF REVIEWS & ITS RELATION WITH PRICE

There are so many factors which contributes towards the price of a listing on AirBnB.While, we already have few conclusions for relationship between various factors and their dependency on prices of a listing,lets analyze if price of a listing dependent upon number of reviews and if yes, how does it varies?

```
reviewsDF = pd.read_csv("reviews.csv")
reviewsDF = reviewsDF.dropna()
```

To retrieve the 'sentiment' of comments - 'positive','negative' or 'neutral' I am using built-in analyzer in the NLTK Python library to assign polarity scoore to each comment.I have assigned Ploarity score to every comment which includes detail like Positivity, negativity neutral and compound.

listing_id	id	reviewer_id	comments	polarity_value	neg	pos	neu	compound
1178162	4724140	4298113	My stay at islam's place was really cool! Good...	{'neg': 0.0, 'neu': 0.648, 'pos': 0.352, 'comp...	0.0	0.352	0.648	0.9626
1178162	4869189	6452964	Great location for both airport and city - gre...	{'neg': 0.0, 'neu': 0.639, 'pos': 0.361, 'comp...	0.0	0.361	0.639	0.9061

Data can be accessed from here

Our data presented in the dataframe consists of comments in various language while we are concerned only with comments in English language.In the next section I am removing all those rows which consists of comments in different language than that of English using langdetect library.

```
def detect_lang(sente):
    sente=str(sente)
    try:
        return detect(sente)
    except:
        return "None"


#taking rows whose language is English
EngReviewsDF=reviewsDF[reviewsDF.language=='en']
```

listing_id	id	reviewer_id	comments	polarity_value	neg	pos	neu	compound	language
1178162	4724140	4298113	My stay at islam's place was really cool! Good...	{'neg': 0.0, 'neu': 0.648, 'pos': 0.352, 'comp...	0.0	0.352	0.648	0.9626	en

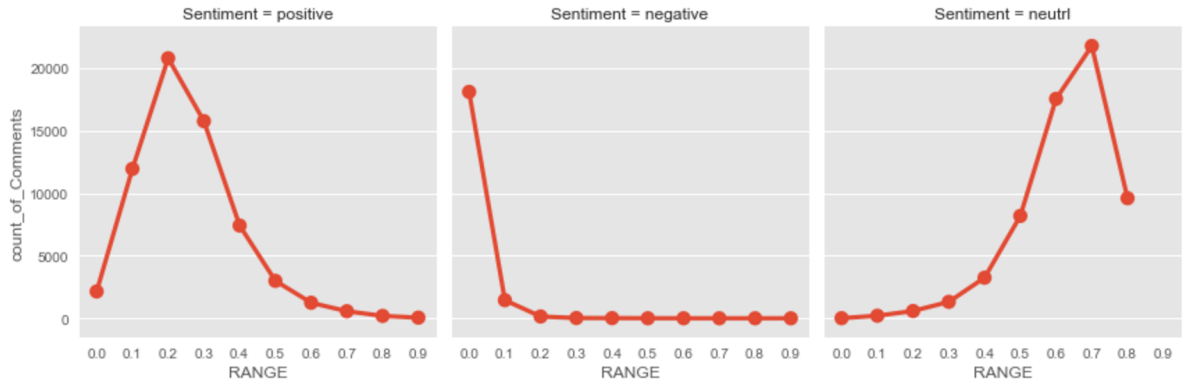
I created graphs to visualize the positive, negative anf neutral score by dividing thernage from 0.0 to 1.0 and counting the reviews between this range.

```
polarDF=EngReviewsDF[['pos']]
polarDF=polarDF.groupby(pd.cut(polarDF["pos"], np.arange(0, 1.1, 0.1))).count()
polarDFneg=polarDFneg.groupby(pd.cut(polarDFneg["neg"], np.arange(0, 1.1, 0.1))).count()
polarDFnut=polarDFnut.groupby(pd.cut(polarDFnut["neu"], np.arange(0, 1.0, 0.1))).count()
```

count_of_Comments	RANGE	Sentiment

count_of_Comments	RANGE	Sentiment
2143	0.0	positive
11940	0.1	positive
18092	0.0	negative
1447	0.1	negative
9	0.0	neutrl
204	0.1	neutrl

Plotted a factorgraph to undersatnd and compare the sentiments of comments travellers mentioned on the listings.

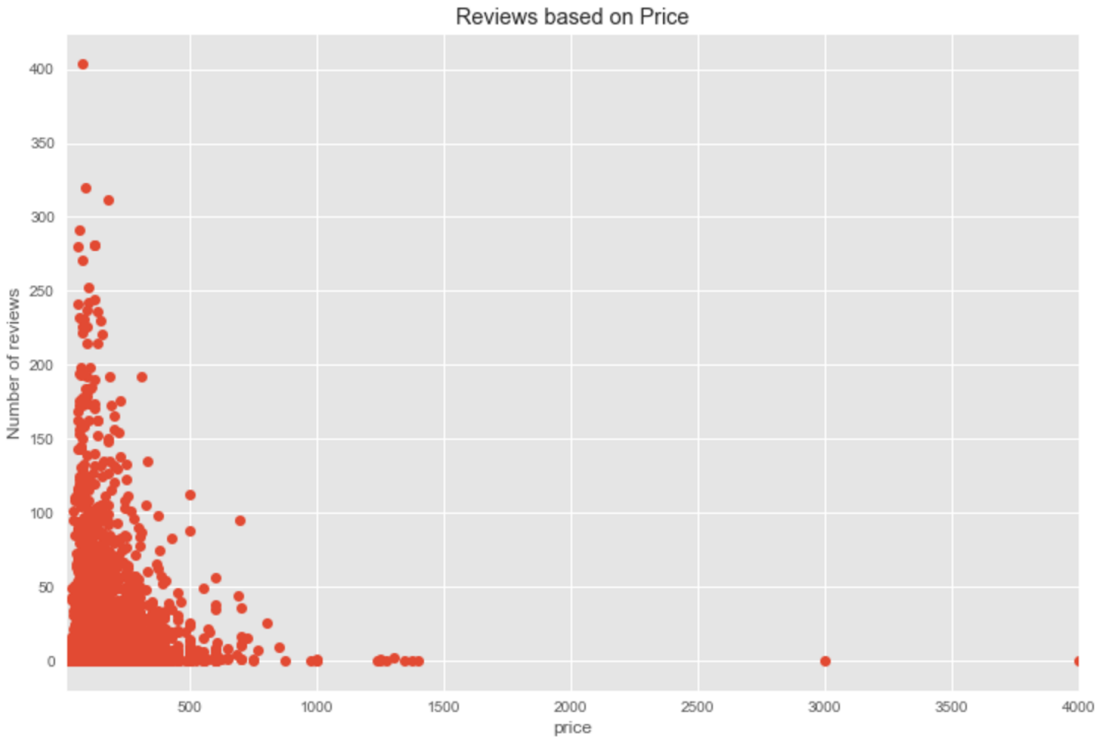


1st Data Point:

It can be seen that Almost none of the texts are classified as having significant amounts of negativity. In fact, a significant amount of them are given exactly 0.0 negativity.It can be clearly seen that most of the comments are neutral.However, a signifigant amount of comments are positive.

We can loosely interpret number of reviews as times people have stayed in the said listing. Of course, this depends on when the listing appeared, how long it appeared for, and various other factors. But it could serve to be useful information to see correlation between price and number of reviews.Lets check if there is any relationship between number of reviews and price of listing.

```
price_review = inputDF[['number_of_reviews', 'price']].sort_values(by = 'price')
```



2nd Data Point:

The graph shows that listings with prices that range around 100 - 400 get the most reviews, probably because they are in the most reasonable price range. The number quickly declines as the price goes up. This indicates that more people book listings that are around $100 - 400 in prices. This shows that it is not necessary for an expensive listing to have large number of reviews.Heneforth there is no exact relation between Prices and Number of Reviews for a listing.

Lets analyze what were the most talked about words in all the comments.

```
for index,row in EngReviewsDF.iterrows():
    words += row['comments']
words_only = [''.join(c for c in s if c not in string.punctuation if c not in nums if c not in ignoreChar) for s in
reviews_data]
comments_filtered_data = ' '.join([word.lower() for word in comments_filtered_data.split() if word not in
cachedStopWords])
```

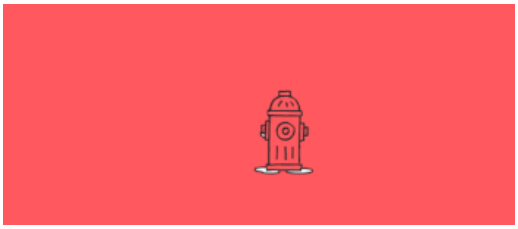
```
wordcloud = WordCloud(width = 1000, height = 700).generate(comments_filtered_data)
```

It can be seen that the most talked about words are "great location", "great host", "walking distance" and "highly recommended".All these reviews and comments plays a big role in attracting the attention of travellers and if there are comments such as "highly recommended" then travellers surely takes a look at the listing.

Conlcusions:

1. It can be clearly seen that most of the comments are neutral and a signifigant amount of comments are positive and almost none of the texts are classified as having significant amounts of negativity.
2. Its clearly visible that listings with prices that range around 100 - 400 get the most reviews, probably because they are in the most reasonable price range. The number quickly declines as the price goes up.
3. the most talked about words are "great location", "great host", "walking distance" and "highly recommended".All these reviews and comments plays a big role in attracting the attention of travellers

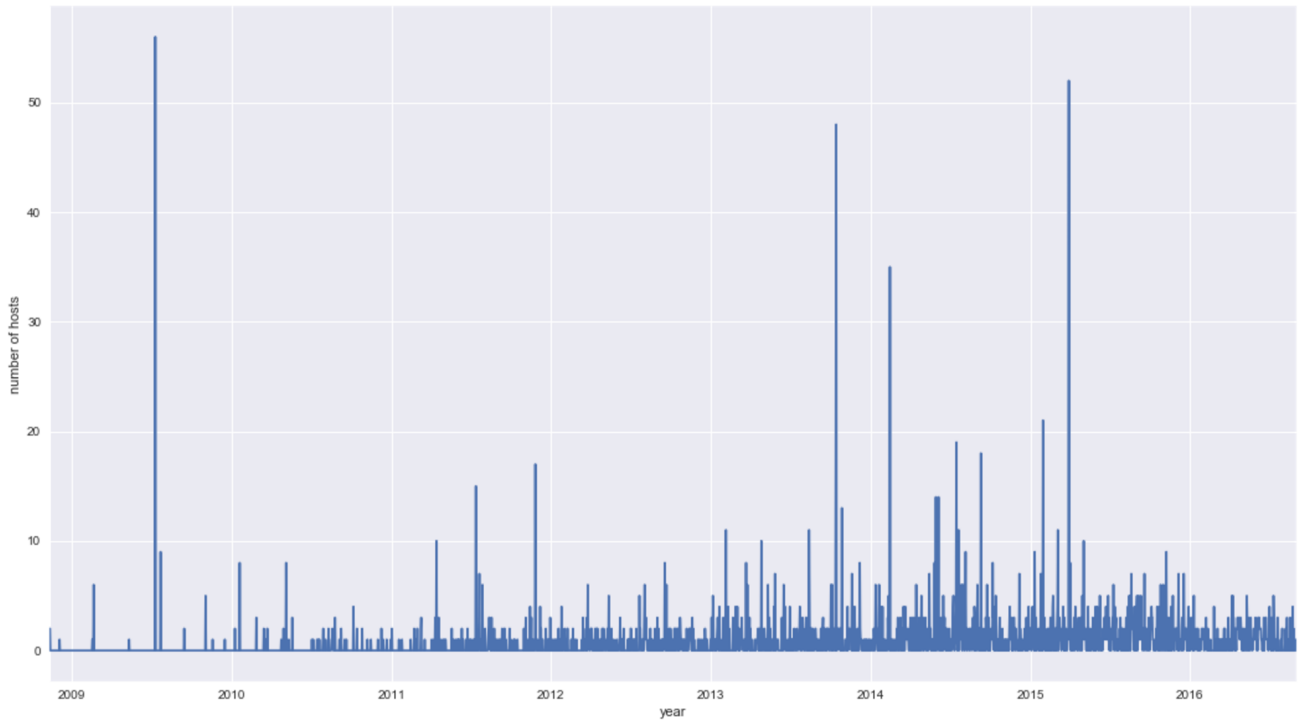
ANALYSIS - 5



HOST ANALYSIS & RECOMMENDATION SYSTEM FOR PRICES

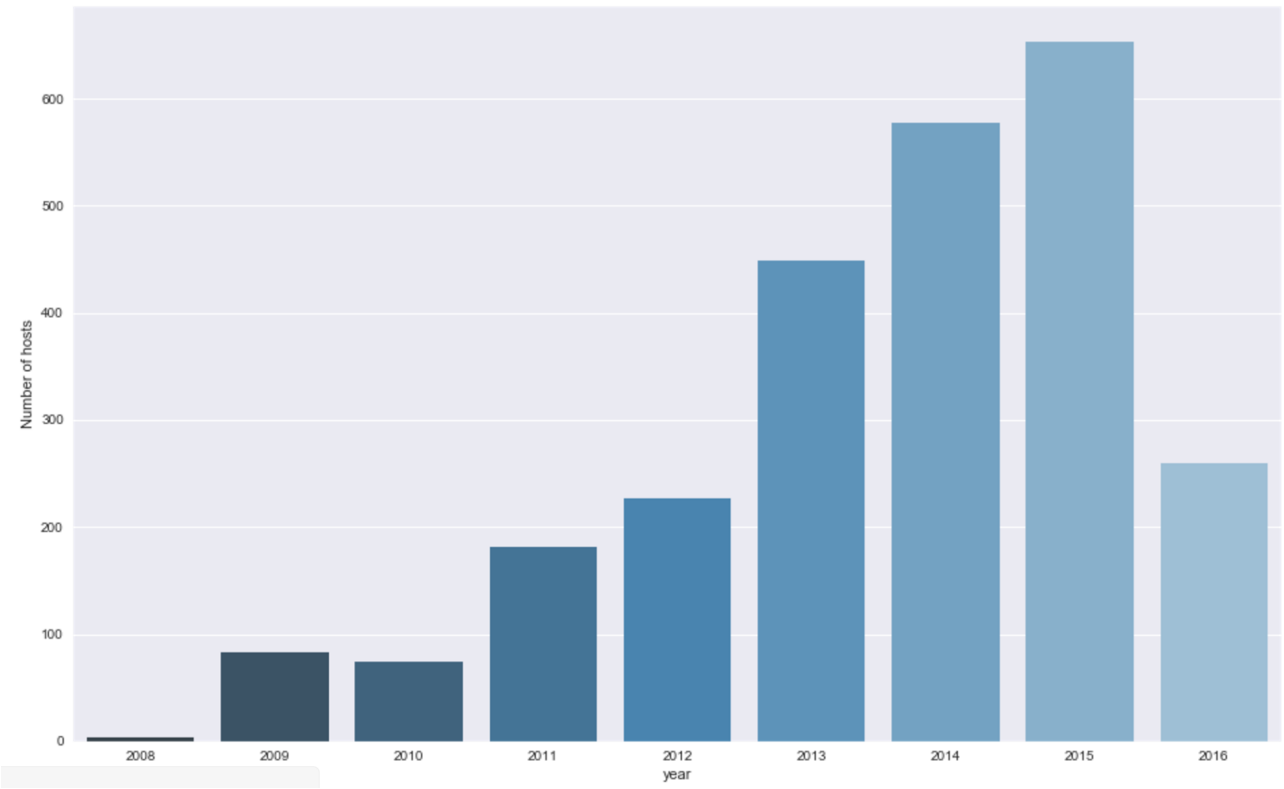
Lets analyze the trend of people hosting AirBnBs in Boston.We can get insights about how much more popular is AirBnB among hosts now than it was two or three years ago. To analyze the same I cleaned the data and separated the host_since date in various columns such as day, date and year and plotted a graph for number of host registered vs each day of the year

```
join_dates = pd.to_datetime(inputDF['host_since']).value_counts().resample('D').mean().fillna(0)
```



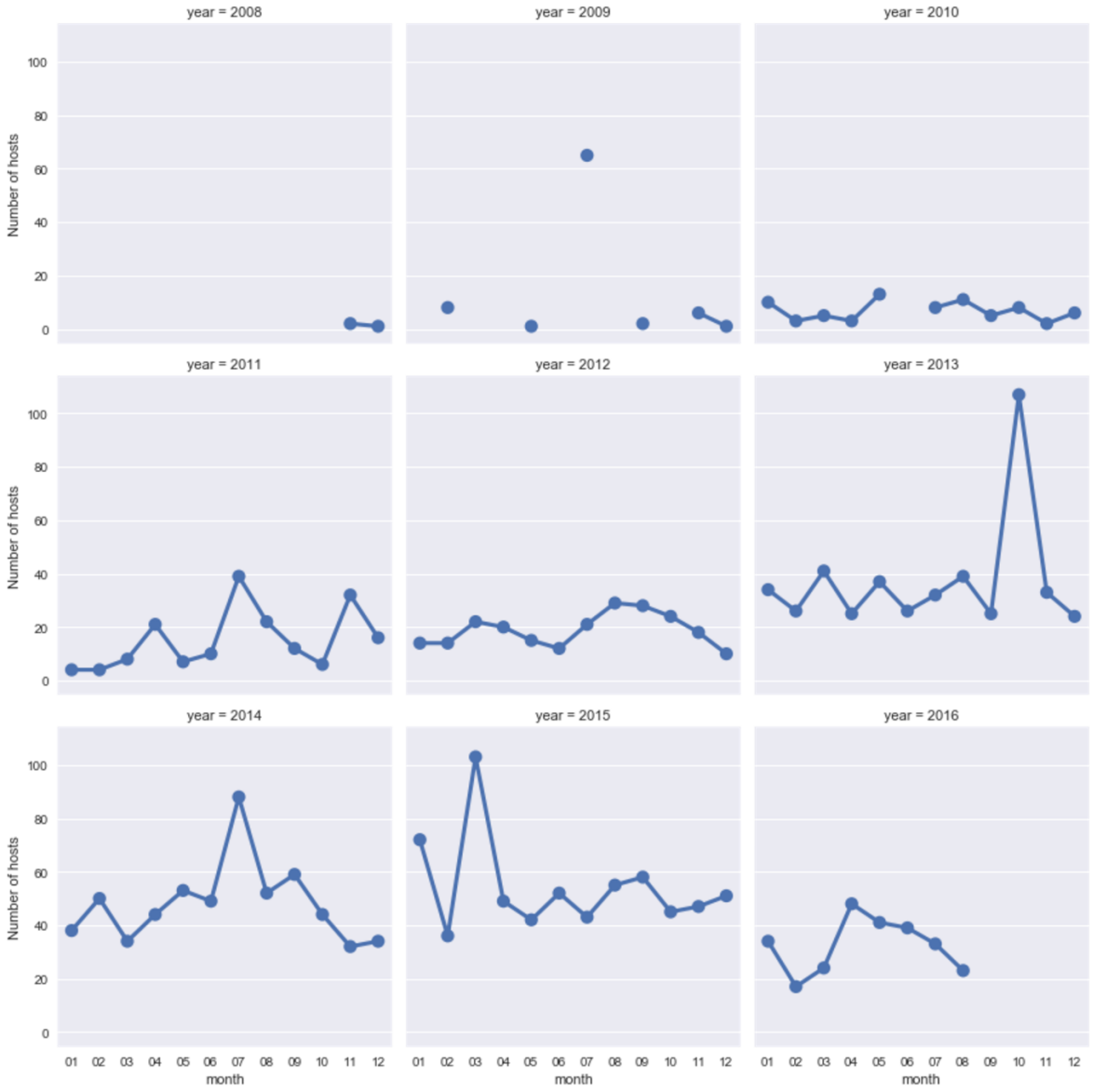
I have plotted the same data for each year.

```
yearDF=inputDF.groupby(['year']).id.count()
```



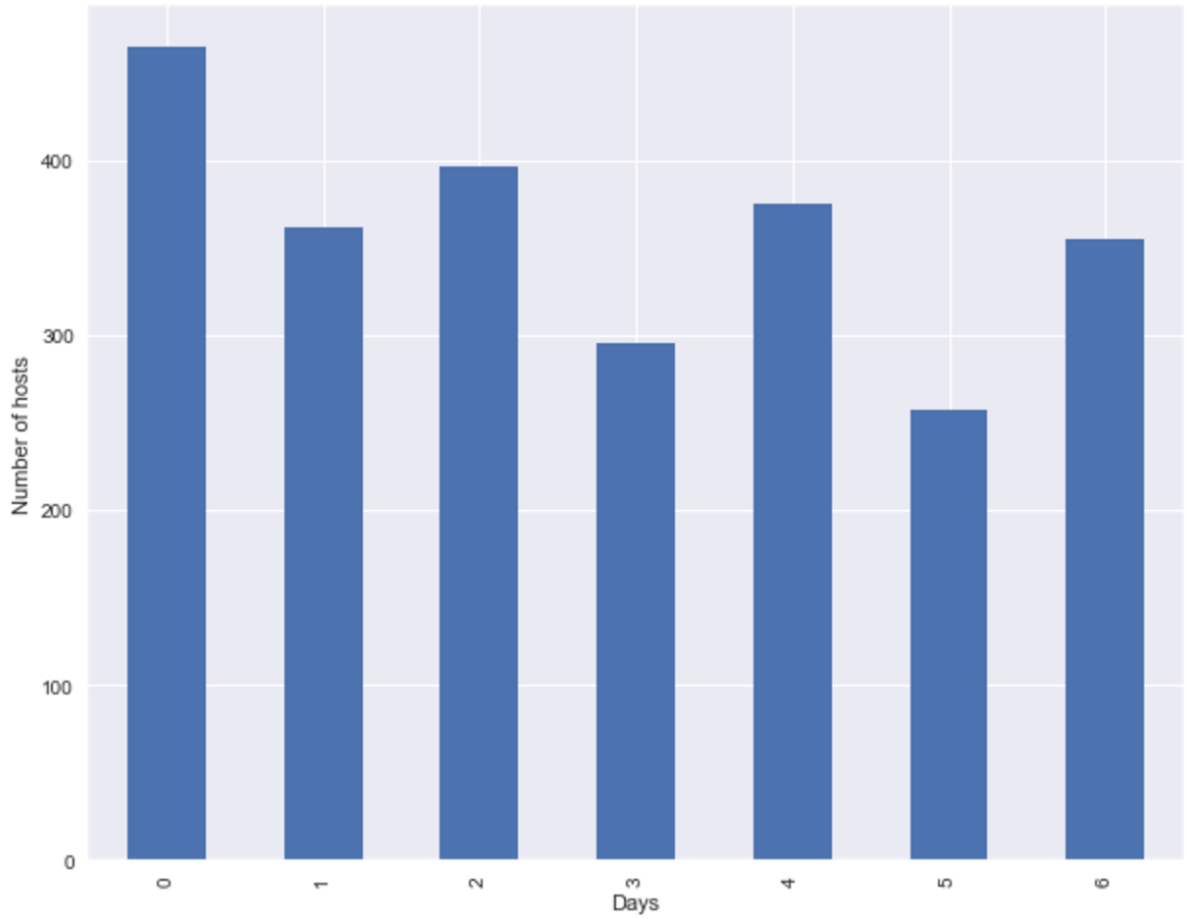
It can be analyzed that Most of the hosts registered in the year 2015.Lets analyze the months

```
yearMonth=inputDF.groupby(['year','month']).id.count()
yearMonth=yearMonth.sort_values(['year','month'],ascending=[1,1])
```



it can be seen that most of the hosts registered in the month of july,november and march.Although there is no specific trend followed.Lets check do people register more on weekends? I have plotted the same where 0 corresponds for mondays and 6 corresponds to sunday

```
pd.to_datetime(inputDF['host_since']).dt.dayofweek.value_counts().sort_index().plot(kind='bar')
```



It shows that people tend to register more on Mondays than any other day of the week.

Recommendation System

After analyzing and looking at all the factors which contributes towards deciding the price of a listing I ma trying to develop a recommendation system for determining the price of test data while training the system on the training data.

I am only considering the columns on which Prices are found to be dependent as the original dataset consists of so many unwanted columns.I have cleaned the dataset by removing arows 0 as their number of bedrooms, batrooms ,beds and converted the price column data into a float to perform operations on the same. Data is presented in categorial form.Hence I am using pandas' get_dummies function to convert Categorial variables into indicator variables.

```
#this will create 4 columns namely flexible, moderate, strict, super_strict_30 which are its categories

cancel_policy = pd.get_dummies(inputDF.cancellation_policy).astype(int)
cancel_policy.head()
```

flexible	moderate	strict	super_strict_30
0	1	0	0
0	1	0	0

```
# Similiarly converting remaining categorial column,instant_booking  into indiactor variables

instant_booking = pd.get_dummies(inputDF.instant_bookable, prefix = 'instant_booking').astype(int)
room_type = pd.get_dummies(inputDF.room_type).astype(int)
```

Now when I have all the categorial data turned into indicator variables,original columns can be dropped and can be replaced by new columns containing indicator variables for the same.

```
# drop original columns and replace them with indicator columns

inputDF = inputDF.drop(['cancellation_policy', 'instant_bookable', 'room_type'], axis = 1)
inputDF = pd.concat((inputDF, cancel_policy, instant_booking, room_type), axis = 1)
```

I have splitted all the amenities and counted the number of amenities and replaced amenities in teh dataframe by this column.

Now after converting all our data into indicator variables we can separateour data into test and train sets using scikit learn's train_test_split function

```
split_data= inputDF.drop(['price'],axis=1)
train1,test1,train2,test2=cross_validation.train_test_split(split_data,inputDF.price, test_size=0.4,train_size =
0.6,random_state=13)



# Lets analyze if linear regression can predict the prices accurately
# mean of prices
mean = np.mean(inputDF.price)

# standard deviation to compare
std = np.std(inputDF.price)

print("mean: " + str(mean))
print ("standard deviation: " + str(std))
```

mean: 168.4856344772546 standard deviation: 117.47652969451681

```
# linear regression testing
linear_reg = linear_model.LinearRegression()
linear_reg.fit(train1, train2)
linear_reg_error = metrics.median_absolute_error(test2, linear_reg.predict(test1))
print ("Linear Regression: " + str(linear_reg_error))
```

Linear Regression: 34.7018093101

This is a small recommendation system which I built using linear Regression.It is not accurate and gives an error of $34.70 in price.The future scope of this project could be using different algorithm to predict accurate pricesof test data.

Addtional Instructions to Run the code

install mentioned libraries

- conda install bokeh
- pip install holidays
- pip install langdetect
- pip install wordcloud and python setup.py install