In [1]:

```
#Importing Key libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
#Brief
# Assumed the role of a  Performance Analyst for AirBnB platfrom that allows individuals
# Over the years AirBnB has grown in popularity and become a key focus of regulations ai
# Hence,the assignment is to analyze Paris Listings  with a focus in pricing.
# And proide leadership with a visul summary of factors affecting pricing and whether th
# impacted listings in the Paris market
```

In [3]:

```
# OBJECTIVES
# 1.Explore and profile the data to correct any uality issues
# 2.Prepare and reform the data for visuazlization
# 3.Visualize the data and give key recommendatins
```

In [4]:

```
#Importing the dataset
Listings = pd.read_csv('C:\\Users\\hp\\Desktop\\Personal Projects\\Listings.csv',encodin
Listings
```

Out[4]:

| | listing_id | name | host_id | host_since | host_location | host_response_time | host_response_ |
|---|---|---|---|---|---|---|---|
| **0** | 281420 | Beautiful Flat in le Village Montmartre, Paris | 1466919 | 2011-12-03 | Paris, Ile-de-France, France | NaN | N |
| **1** | 3705183 | 39 mÃ□Â² Paris (Sacre CÃ□â□□ur) | 10328771 | 2013-11-29 | Paris, Ile-de-France, France | NaN | N |
| **2** | 4082273 | Lovely apartment with Terrace, 60m2 | 19252768 | 2014-07-31 | Paris, Ile-de-France, France | NaN | N |
| **3** | 4797344 | Cosy studio (close to Eiffel tower) | 10668311 | 2013-12-17 | Paris, Ile-de-France, France | NaN | N |
| **4** | 4823489 | Close to Eiffel Tower - Beautiful flat : 2 rooms | 24837558 | 2014-12-14 | Paris, Ile-de-France, France | NaN | N |
| **...** | ... | ... | ... | ... | ... | ... | |
| **279707** | 38338635 | Appartement T2 neuf prÃ□Â¨s du tram T3a Porte ... | 31161181 | 2015-04-13 | Paris, Ile-de-France, France | NaN | N |
| **279708** | 38538692 | Cozy Studio in Montmartre | 10294858 | 2013-11-27 | Paris, Ile-de-France, France | NaN | N |
| **279709** | 38683356 | Nice and cosy mini-appartement in Paris | 2238502 | 2012-04-27 | Paris, Ile-de-France, France | NaN | N |
| **279710** | 39659000 | Charming apartment near Rue Saint Maur / Oberk... | 38633695 | 2015-07-16 | Paris, Ile-de-France, France | NaN | N |
| **279711** | 40219504 | Cosy apartment with view on Canal St Martin | 6955618 | 2013-06-17 | Paris, Ile-de-France, France | NaN | N |

279712 rows × 33 columns

In [5]:

```
#Reading the data
Listings
```

Out[5]:

| | listing_id | name | host_id | host_since | host_location | host_response_time | host_response_ |
|---|---|---|---|---|---|---|---|
| **0** | 281420 | Beautiful Flat in le Village Montmartre, Paris | 1466919 | 2011-12-03 | Paris, Ile-de-France, France | NaN | N |
| **1** | 3705183 | 39 mÃ□Â² Paris (Sacre CÃ□â□□ur) | 10328771 | 2013-11-29 | Paris, Ile-de-France, France | NaN | N |
| **2** | 4082273 | Lovely apartment with Terrace, 60m2 | 19252768 | 2014-07-31 | Paris, Ile-de-France, France | NaN | N |
| **3** | 4797344 | Cosy studio (close to Eiffel tower) | 10668311 | 2013-12-17 | Paris, Ile-de-France, France | NaN | N |
| **4** | 4823489 | Close to Eiffel Tower - Beautiful flat : 2 rooms | 24837558 | 2014-12-14 | Paris, Ile-de-France, France | NaN | N |
| **...** | ... | ... | ... | ... | ... | ... | |
| **279707** | 38338635 | Appartement T2 neuf prÃ□Â¨s du tram T3a Porte ... | 31161181 | 2015-04-13 | Paris, Ile-de-France, France | NaN | N |
| **279708** | 38538692 | Cozy Studio in Montmartre | 10294858 | 2013-11-27 | Paris, Ile-de-France, France | NaN | N |
| **279709** | 38683356 | Nice and cosy mini-appartement in Paris | 2238502 | 2012-04-27 | Paris, Ile-de-France, France | NaN | N |
| **279710** | 39659000 | Charming apartment near Rue Saint Maur / Oberk... | 38633695 | 2015-07-16 | Paris, Ile-de-France, France | NaN | N |
| **279711** | 40219504 | Cosy apartment with view on Canal St Martin | 6955618 | 2013-06-17 | Paris, Ile-de-France, France | NaN | N |

279712 rows × 33 columns

In [6]:

```
#Accessing the column names
column_names = list(Listings.columns)
print(column_names)
```

```
['listing_id', 'name', 'host_id', 'host_since', 'host_location', 'host_response_time',
'host_response_rate', 'host_acceptance_rate', 'host_is_superhost', 'host_total_listings_
count', 'host_has_profile_pic', 'host_identity_verified', 'neighbourhood', 'district',
'city', 'latitude', 'longitude', 'property_type', 'room_type', 'accommodates', 'bedroom
s', 'amenities', 'price', 'minimum_nights', 'maximum_nights', 'review_scores_rating', 'r
eview_scores_accuracy', 'review_scores_cleanliness', 'review_scores_checkin', 'review_sc
ores_communication', 'review_scores_location', 'review_scores_value', 'instant_bookabl
e']
```

In [7]:

```
#Subsetting the 'host_since' date column
Host_since = Listings['host_since']
Host_since
#This is for the purpose of checking the datatype of host_since column
```

Out[7]:

```
0         2011-12-03
1         2013-11-29
2         2014-07-31
3         2013-12-17
4         2014-12-14
             ...
279707    2015-04-13
279708    2013-11-27
279709    2012-04-27
279710    2015-07-16
279711    2013-06-17
Name: host_since, Length: 279712, dtype: object
```

In [8]:

```
#Changing the formart of the column type to datetime formart
Listings['host_since'] = pd.to_datetime(Listings['host_since'])
Listings['host_since']
```

Out[8]:

```
0        2011-12-03
1        2013-11-29
2        2014-07-31
3        2013-12-17
4        2014-12-14
            ...
279707   2015-04-13
279708   2013-11-27
279709   2012-04-27
279710   2015-07-16
279711   2013-06-17
Name: host_since, Length: 279712, dtype: datetime64[ns]
```

In [9]:

```
City_Listings = Listings['city'].unique()
City_Listings
#To identify the different cities in this dataset
```

Out[9]:

```
array(['Paris', 'New York', 'Bangkok', 'Rio de Janeiro', 'Sydney',
       'Istanbul', 'Rome', 'Hong Kong', 'Mexico City', 'Cape Town'],
      dtype=object)
```

In [10]:

```
Paris_Listings = Listings[Listings['city']=='Paris'][['city','host_since','neighbourhood
Paris_Listings
#To subset Paris listings only with time from when dwellers took up the space,their
#capacity of the space and price.
```

Out[10]:

| | city | host_since | neighbourhood | accommodates | price |
|---|---|---|---|---|---|
| **0** | Paris | 2011-12-03 | Buttes-Montmartre | 2 | 53 |
| **1** | Paris | 2013-11-29 | Buttes-Montmartre | 2 | 120 |
| **2** | Paris | 2014-07-31 | Elysee | 2 | 89 |
| **3** | Paris | 2013-12-17 | Vaugirard | 2 | 58 |
| **4** | Paris | 2014-12-14 | Passy | 2 | 60 |
| **...** | ... | ... | ... | ... | ... |
| **279707** | Paris | 2015-04-13 | Observatoire | 2 | 120 |
| **279708** | Paris | 2013-11-27 | Buttes-Montmartre | 2 | 60 |
| **279709** | Paris | 2012-04-27 | Buttes-Montmartre | 2 | 50 |
| **279710** | Paris | 2015-07-16 | Popincourt | 2 | 105 |
| **279711** | Paris | 2013-06-17 | Enclos-St-Laurent | 2 | 70 |

64690 rows × 5 columns

In [11]:

```
#To check only Paris city was filtered
Only_Paris_Listings = Paris_Listings['city'].unique()
Only_Paris_Listings
```

Out[11]:

```
array(['Paris'], dtype=object)
```

In [12]:

```
#To check for missing data
Missing_data = Paris_Listings.isnull().sum()
Missing_data
```

Out[12]:

```
city              0
host_since       33
neighbourhood     0
accommodates      0
price             0
dtype: int64
```

In [13]:

```
# Droppping rows with missing values
New_Paris_Listings = Paris_Listings.dropna()
New_Paris_Listings
```

Out[13]:

| | city | host_since | neighbourhood | accommodates | price |
|---|---|---|---|---|---|
| **0** | Paris | 2011-12-03 | Buttes-Montmartre | 2 | 53 |
| **1** | Paris | 2013-11-29 | Buttes-Montmartre | 2 | 120 |
| **2** | Paris | 2014-07-31 | Elysee | 2 | 89 |
| **3** | Paris | 2013-12-17 | Vaugirard | 2 | 58 |
| **4** | Paris | 2014-12-14 | Passy | 2 | 60 |
| **...** | ... | ... | ... | ... | ... |
| **279707** | Paris | 2015-04-13 | Observatoire | 2 | 120 |
| **279708** | Paris | 2013-11-27 | Buttes-Montmartre | 2 | 60 |
| **279709** | Paris | 2012-04-27 | Buttes-Montmartre | 2 | 50 |
| **279710** | Paris | 2015-07-16 | Popincourt | 2 | 105 |
| **279711** | Paris | 2013-06-17 | Enclos-St-Laurent | 2 | 70 |

64657 rows × 5 columns

In [14]:

```
New_Paris_Listings.describe()
#To obatin statistical summaries of numeric columns
```

Out[14]:

| | accommodates | price |
|---|---|---|
| **count** | 64657.000000 | 64657.000000 |
| **mean** | 3.037877 | 113.104614 |
| **std** | 1.588382 | 214.479626 |
| **min** | 0.000000 | 0.000000 |
| **25%** | 2.000000 | 59.000000 |
| **50%** | 2.000000 | 80.000000 |
| **75%** | 4.000000 | 120.000000 |
| **max** | 16.000000 | 12000.000000 |

In [15]:

```
Paris_listings_neighbourhood = New_Paris_Listings.groupby('neighbourhood')['price'].mean
Paris_listings_neighbourhood
#To check pricing per neighbourhood,sorted from most expensive neighbourhood to the leas
```

Out[15]:

```
neighbourhood
Elysee                 210.536765
Louvre                 175.379972
Passy                  161.190476
Palais-Bourbon         156.891525
Luxembourg             155.638639
Bourse                 149.496801
Hotel-de-Ville         144.515228
Temple                 138.429300
Pantheon               122.696120
Opera                  119.050713
Vaugirard              106.842073
Enclos-St-Laurent      102.988752
Batignolles-Monceau    102.615616
Observatoire           101.873591
Gobelins                98.110184
Popincourt              90.518955
Reuilly                 89.058402
Buttes-Montmartre       87.222069
Buttes-Chaumont         82.690182
Menilmontant            74.911561
Name: price, dtype: float64
```

In [16]:

```
Paris_listings_accomodtions = New_Paris_Listings[New_Paris_Listings['neighbourhood'] ==
Paris_listings_accomodtions
#To check the accomodation rates of Elysee estate-the most expensive neighbourhood
```

Out[16]:

```
accommodates
14    971.000000
13    842.500000
11    805.000000
16    800.000000
12    529.625000
10    500.857143
9     440.272727
7     411.538462
8     405.518519
6     355.508571
5     328.817073
4     212.096070
2     155.103352
3     152.828767
1      79.522222
0       0.000000
Name: price, dtype: float64
```

In [17]:

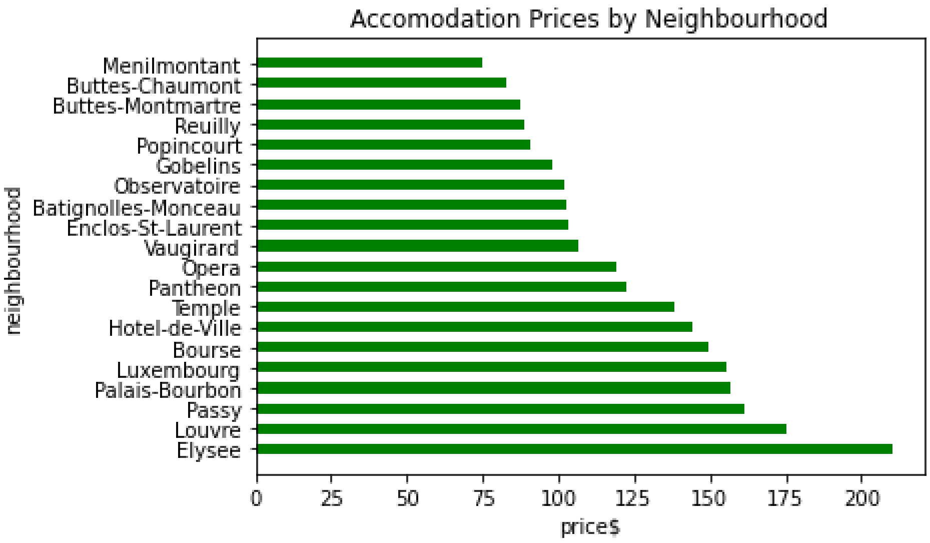
```
Paris_listings_overtime = New_Paris_Listings.groupby(New_Paris_Listings['host_since'].dt
Paris_listings_overtime
# To check the hosting trend from when AirBnB began listig in Paris
```

Out[17]:

| | price | host_since |
|---|---|---|
| **host_since** | | |
| **2008** | 77.750000 | 4 |
| **2009** | 159.641509 | 106 |
| **2010** | 125.031250 | 416 |
| **2011** | 124.828230 | 1339 |
| **2012** | 111.578615 | 4592 |
| **2013** | 107.096414 | 8142 |
| **2014** | 100.253800 | 10922 |
| **2015** | 103.646250 | 12147 |
| **2016** | 114.159847 | 8871 |
| **2017** | 108.658888 | 4585 |
| **2018** | 138.209362 | 4294 |
| **2019** | 129.757113 | 5694 |
| **2020** | 141.456038 | 3412 |
| **2021** | 93.488722 | 133 |

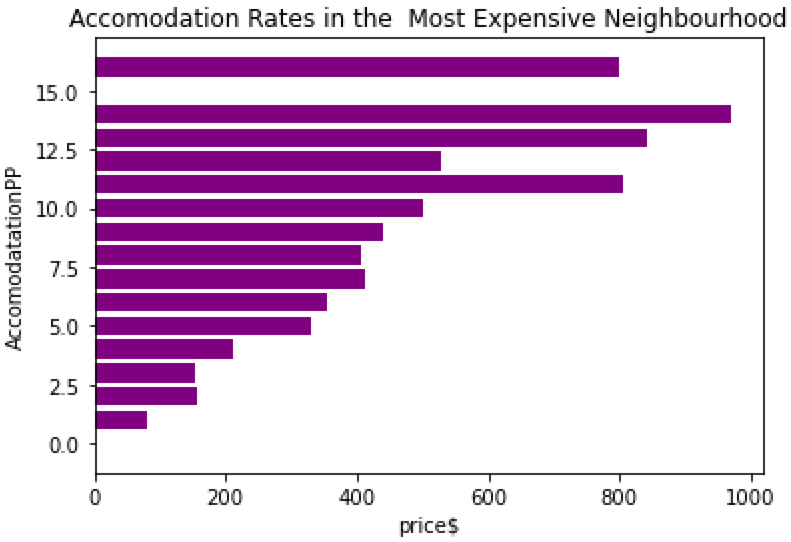
In [18]:

```
plt.barh(Paris_listings_neighbourhood.index,Paris_listings_neighbourhood.values,color =
plt.title('Accomodation Prices by Neighbourhood')
plt.ylabel('neighbourhood')
plt.xlabel('price' + ('$'))
plt.show()
#A visual summary of the Accomodtaion pricing against neighbourhoods
```

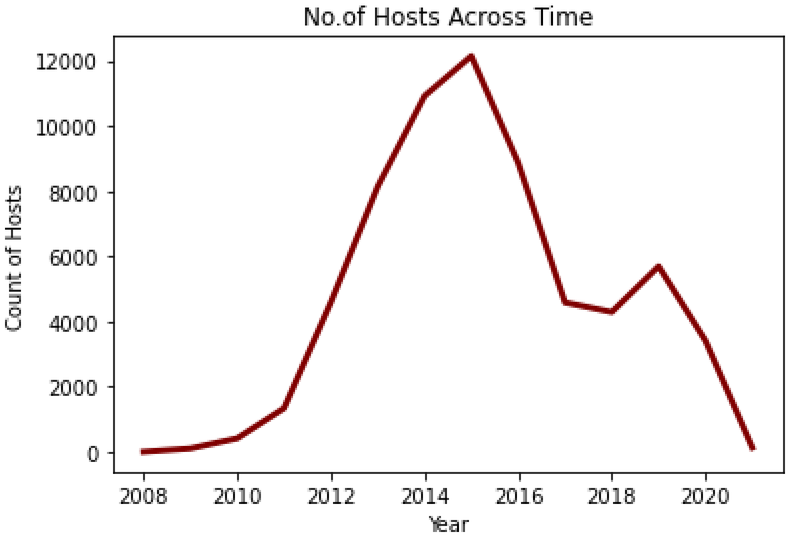


In [19]:

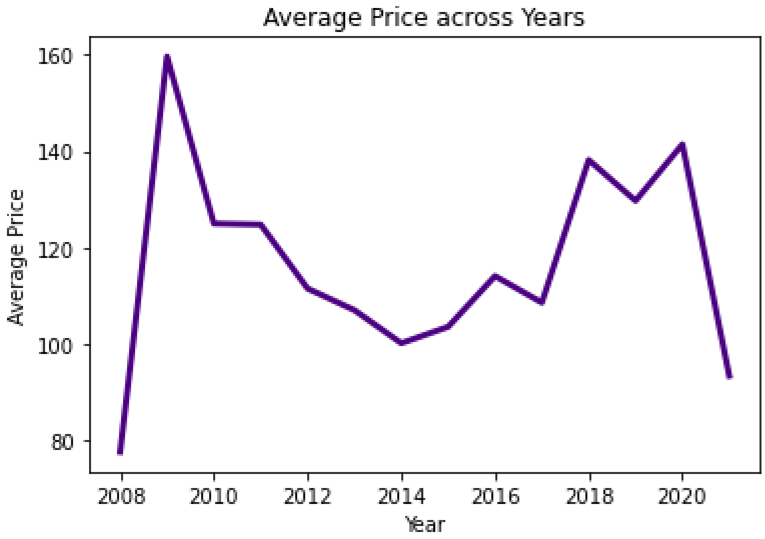
```
plt.barh(Paris_listings_accomodtions.index,Paris_listings_accomodtions.values, colo
plt.title('Accomodation Rates in the  Most Expensive Neighbourhood')
plt.ylabel('Accomodatation' + ('PP'))
plt.xlabel('price' + ('$'))
plt.show()
#A visual presentation of of Accomodation capacity with rates
```



```
In [20]: plt.plot(Paris_listings_overtime['host_since'], linestyle = 'solid',color ='maroon',line
         plt.ylabel('Count of Hosts')
         plt.xlabel('Year')
         plt.title('No.of Hosts Across Time')
         plt.show()
         #A trendline of number of hosts overtime
```

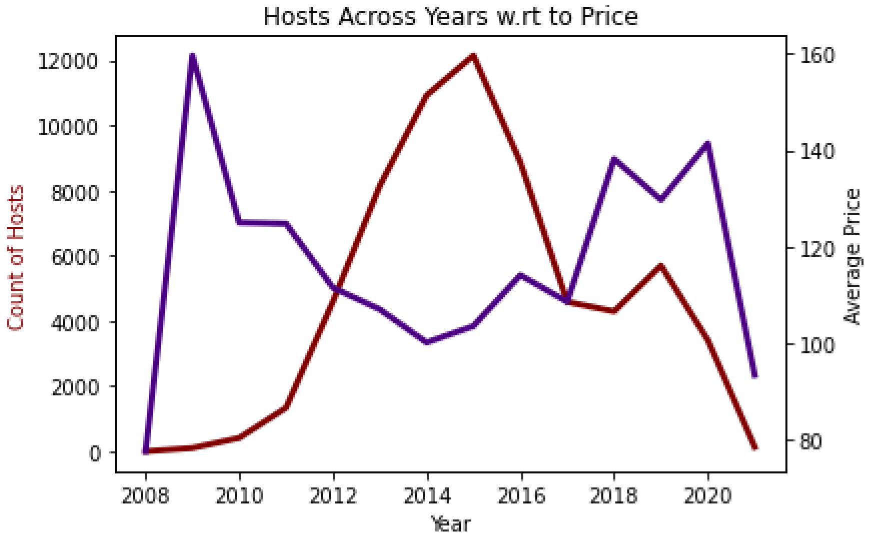


```
In [21]: plt.plot(Paris_listings_overtime['price'], linestyle = 'solid', linewidth = '3',color =
         plt.ylabel('Average Price')
         plt.xlabel('Year')
         plt.title('Average Price across Years')
         plt.show()
         #A trend representation of Pricing across Years
```



In [22]:

```
fig, ax1 = plt.subplots()
ax1.plot(Paris_listings_overtime['host_since'], linestyle = 'solid',color ='maroon',line
ax1.set_ylabel('Count of Hosts',color = 'maroon')
ax1.set_xlabel('Year')
ax2 = ax1.twinx()
ax2.plot(Paris_listings_overtime['price'], linestyle = 'solid', linewidth = '3',color =
ax2.set_ylabel('Average Price')
plt.title('Hosts Across Years w.rt to Price')
plt.show()
#An interaction of number of hosts overtime with Pricing
```



In [23]:

```
# CONCLUSION
# The 2015 regulations impacted the Paris market with steep drop of travellers opting fo
# For pricing there are no clear reasons for price behaviour,other than to make the assu
# within Paris due to its assumed posh and luxurious status.
# Hence is uniquely not impacted with a reduction of travellers in need of AirBnB space
```