Capstone Project Creation

IBM SkillsBuild Europe Delivery - Data Analytics

Pre-requisite

- Understanding of Python, Power BI or Tableau
- Understanding of Data Cleaning
- Understanding Data Visualization

Level of Exercise: Intermediate

Duration: approximately 3 hours

Data Analytics of Airbnb Data:

Objective:

In this exericise, you will be performing Data Analytics on an Open Dataset dataset coming from Airbnb. Some of the tasks include

- Data Cleaning.
- Data Transformation
- Data Visualization.

Overview of Airbnb Data:

People's main criteria when visiting new places are reasonable accommodation and food. Airbnb (Air-Bed-Breakfast) is an online marketplace created to meet this need of people by renting out their homes for a short term. They offer this facility at a relatively lower price than hotels. Further people worldwide prefer the homely and economical service offered by them. They offer services across various geographical locations

Dataset Source

YOu can get the dataset for this assessment using the following link:
https://www.kaggle.com/datasets/arianazmoudeh/airbnbopendata

This dataset contains information such as the neighborhood offering these services, room type, price,avaliabilty, reviews, service fee, cancellation policy and rules to use the house. This analysis will help airbnb in improving its services.

So all the best for your Data Analytics Journey on Airbnb data!!!

Task 1: Data Loading (Python)

1. Read the csv file and load it into a pandas dataframe.
2. Display the first five rows of your dataframe.
3. data types of the columns.

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

In [2]:
```
## Read the csv file
pj = pd.read_csv("C:\\Users\\User\\Desktop\\Airbnb_Open_Data.csv", low_memory=False)
```

In [3]:
```
##display the result of output
pj
```

Out[3]:

	id	NAME	host id	host_identity_verified	host name	neighbourhood group	neighbourhoo
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn	Kensingto
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan	Midtow
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan	Harle
3	1002755	NaN	85098326012	unconfirmed	Garry	Brooklyn	Clinton H
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan	East Harle
...	...	...	...	...	...	...	
102594	6092437	Spare room in Williamsburg	12312296767	verified	Krik	Brooklyn	Williamsbu
102595	6092990	Best Location near Columbia U	77864383453	unconfirmed	Mifan	Manhattan	Morningsid Heigh
102596	6093542	Comfy, bright room in Brooklyn	69050334417	unconfirmed	Megan	Brooklyn	Park Slop
102597	6094094	Big Studio-One Stop from Midtown	11160591270	unconfirmed	Christopher	Queens	Long Island Ci
102598	6094647	585 sf Luxury Studio	68170633372	unconfirmed	Rebecca	Manhattan	Upper We Sid

102599 rows × 26 columns

In [4]:
```
## Display the first 5 rows
```

Out[4]:

	id	NAME	host id	host_identity_verified	host name	neighbourhood group	neighbourhood	
0	1001254	Clean & quiet apt home by the park	80014485718	unconfirmed	Madaline	Brooklyn	Kensington	40.64
1	1002102	Skylit Midtown Castle	52335172823	verified	Jenna	Manhattan	Midtown	40.75
2	1002403	THE VILLAGE OF HARLEM....NEW YORK !	78829239556	NaN	Elise	Manhattan	Harlem	40.80
3	1002755	NaN	85098326012	unconfirmed	Garry	Brooklyn	Clinton Hill	40.68
4	1003689	Entire Apt: Spacious Studio/Loft by central park	92037596077	verified	Lyndon	Manhattan	East Harlem	40.79

5 rows × 26 columns

In [5]:
```
## Display the data types
pj.dtypes
```

Out[5]:
```
id                                int64
NAME                             object
host id                           int64
host_identity_verified           object
host name                        object
neighbourhood group              object
neighbourhood                    object
lat                             float64
long                            float64
country                          object
country code                     object
instant_bookable                 object
cancellation_policy              object
room type                        object
Construction year               float64
price                            object
service fee                      object
minimum nights                  float64
number of reviews               float64
last review                      object
reviews per month               float64
review rate number              float64
calculated host listings count  float64
availability 365                float64
house_rules                      object
license                          object
dtype: object
```

Task 2a: Data Cleaning (Any Tool)

1. Drop some of the unwanted columns. These include `host id`, `id`, `country` and `country code` from the dataset.
2. State the reason for not including these columns for your Data Analytics.

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots before and after the elimination of the columns.

In [6]:
```
## Dropping unwanted columns
pj.drop(columns = ['host id', 'id', 'country', 'country code'], axis = 1, inplace = True
```

In [7]:
```
## displaying the results after dropping the columns
pj
```

Out[7]:

	NAME	host_identity_verified	host name	neighbourhood group	neighbourhood	lat	long
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.97237
1	Skylit Midtown Castle	verified	Jenna	Manhattan	Midtown	40.75362	-73.98377
2	THE VILLAGE OF HARLEM....NEW YORK !	NaN	Elise	Manhattan	Harlem	40.80902	-73.94190
3	NaN	unconfirmed	Garry	Brooklyn	Clinton Hill	40.68514	-73.95976
4	Entire Apt: Spacious Studio/Loft by central park	verified	Lyndon	Manhattan	East Harlem	40.79851	-73.94399
...	...	...	...	...	...	...	...
102594	Spare room in Williamsburg	verified	Krik	Brooklyn	Williamsburg	40.70862	-73.94651
102595	Best Location near Columbia U	unconfirmed	Mifan	Manhattan	Morningside Heights	40.80460	-73.96545
102596	Comfy, bright room in Brooklyn	unconfirmed	Megan	Brooklyn	Park Slope	40.67505	-73.98045
102597	Big Studio-One Stop from Midtown	unconfirmed	Christopher	Queens	Long Island City	40.74989	-73.93777
102598	585 sf Luxury Studio	unconfirmed	Rebecca	Manhattan	Upper West Side	40.76807	-73.98342

102599 rows × 22 columns

In [ ]:
```
## stating the reson of dropping the columns
Uniqueness and Irrelevance:

'host id' and 'id' columns often represent unique identifiers or keys for individual lis
'country' and 'country code' might be irrelevant if the dataset pertains to a single cou
Reducing Redundancy and Noise:

Reducing the number of irrelevant or redundant columns helps streamline the dataset and
Focusing on Key Features:

For specific analytics, such as price prediction or neighborhood-based analysis, columns
```

Task 2b: Data Cleaning (Python)

- Check for missing values in the dataframe and display the count in ascending order. **If the values are missing, impute the values as per the datatype of the columns.**
- Check whether there are any duplicate values in the dataframe and, if present, remove them.
- Display the total number of records in the dataframe before and after removing the duplicates.

In [8]:
```
## Check for missing values in the dataframe and display the count in ascending order.

pj.isnull().sum().sort_values(ascending = True)
```

Out[8]:
```
room type                            0
lat                                  8
long                                 8
neighbourhood                       16
neighbourhood group                 29
cancellation_policy                 76
instant_bookable                   105
number of reviews                  183
Construction year                  214
price                              247
NAME                               250
service fee                        273
host_identity_verified             289
calculated host listings count     319
review rate number                 326
host name                          406
minimum nights                     409
availability 365                   448
reviews per month                15879
last review                      15893
house_rules                      52131
license                         102597
dtype: int64
```

In [9]:
```
#dtypes object or int or float

for col in pj.columns:
    if pj[col].dtype == 'O':
        pj[col].fillna(value=pj[col].mode().iloc[0], inplace = True)
    else:
        pj[col].fillna(value=pj[col].median(), inplace = True)
```

In [10]:
```
#dataframe after filling in missing values
pj.head()
```

Out[10]:

	NAME	host_identity_verified	host name	neighbourhood group	neighbourhood	lat	long	instant_
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.97237	
1	Skylit Midtown Castle	verified	Jenna	Manhattan	Midtown	40.75362	-73.98377	
2	THE VILLAGE OF HARLEM....NEW YORK !	unconfirmed	Elise	Manhattan	Harlem	40.80902	-73.94190	
3	Home away from home	unconfirmed	Garry	Brooklyn	Clinton Hill	40.68514	-73.95976	
4	Entire Apt: Spacious Studio/Loft by central park	verified	Lyndon	Manhattan	East Harlem	40.79851	-73.94399	

5 rows × 22 columns

In [11]:
```
## Check whether there are any duplicate values in the dataframe and if present remove t
pj.duplicated().sum()
```

Out[11]:
```
3461
```

In [12]:
```
## Total number of records
pj.shape
```

Out[12]:
```
(102599, 22)
```

In [13]:
```
## Dropping the duplicates
pj.drop_duplicates(inplace = True)
```

In [14]:
```
## Display the total number of records in the dataframe after removing the duplicates.
pj.shape
```

Out[14]:
```
(99138, 22)
```

Task 3: Data Transformation (Any Tool)

- Rename the column `availability 365` to `days_booked`
- Convert all column names to lowercase and replace the spaces in the column names with an underscore "_".
- Remove the dollar sign and comma from the columns `price` and `service_fee`. If necessary, convert these two columns to the appropriate data type.

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [15]:
```
## Rename the column.
pj.rename(columns={'availability 365':  'days_booked'})
```

Out[15]:

	NAME	host_identity_verified	host name	neighbourhood group	neighbourhood	lat	long	ir
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.97237	
1	Skylit Midtown Castle	verified	Jenna	Manhattan	Midtown	40.75362	-73.98377	
2	THE VILLAGE OF HARLEM....NEW YORK !	unconfirmed	Elise	Manhattan	Harlem	40.80902	-73.94190	
3	Home away from home	unconfirmed	Garry	Brooklyn	Clinton Hill	40.68514	-73.95976	
4	Entire Apt: Spacious Studio/Loft by central park	verified	Lyndon	Manhattan	East Harlem	40.79851	-73.94399	
...	...	...	...	...	...	...	...	...
102053	Cozy bright room near Prospect Park	unconfirmed	Mariam	Brooklyn	Flatbush	40.64945	-73.96108	
102054	Private Bedroom with Amazing Rooftop View	verified	Trey	Brooklyn	Bushwick	40.69872	-73.92718	
102055	Pretty Brooklyn One-Bedroom for 2 to 4 people	verified	Michael	Brooklyn	Bedford-Stuyvesant	40.67810	-73.90822	
102056	Room & private bathroom in historic Harlem	unconfirmed	Shireen	Manhattan	Harlem	40.81248	-73.94317	
102057	Rosalee Stewart	verified	Stanley	Manhattan	Harlem	40.81315	-73.94747	

99138 rows × 22 columns

In [16]:
```
## Convert all column names to lowercase and replace the spaces with an underscore "_"
pj.columns = pj.columns.str.lower().str.replace(' ', '_')
```

In [17]:
```
#output
pj.columns
```

Out[17]:
```
Index(['name', 'host_identity_verified', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'lat', 'long', 'instant_bookable',
       'cancellation_policy', 'room_type', 'construction_year', 'price',
       'service_fee', 'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'review_rate_number',
       'calculated_host_listings_count', 'availability_365', 'house_rules',
       'license'],
      dtype='object')
```

In [18]:
```
## Remove the dollar sign and comma from the columns. If necessary, convert these two co
def remove_signs(value):
    return float(value.replace("$","").replace(",",""))
```

In [19]:
```
pj['price'] = pj['price'].apply(lambda x: remove_signs(x))
pj['service_fee'] = pj['service_fee'].apply(lambda x: remove_signs(x))
```

In [20]:
```
## Displaying the output of the price = this shows that the dollar sign has been removed
pj['price']
```

Out[20]:
```
0          966.0
1          142.0
2          620.0
3          368.0
4          204.0
           ...
102053     696.0
102054     909.0
102055     387.0
102056     848.0
102057    1128.0
Name: price, Length: 99138, dtype: float64
```

In [21]:
```
## Displaying the output of service_fee  = this shows that the dollar sign has been remo
pj['service_fee']
```

Out[21]:
```
0         193.0
1          28.0
2         124.0
3          74.0
4          41.0
          ...
102053     41.0
102054     41.0
102055     41.0
102056     41.0
102057     41.0
Name: service_fee, Length: 99138, dtype: float64
```

Task 4: Exploratory Data Analysis (Any Tool)

- List the count of various room types avaliable in the dataset.
- Which room type has the most strict cancellation policy?
- List the average price per neighborhood group, and highlight the most expensive neighborhood to rent from.

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [22]:
```
## List the count of various room types avaliable with Airbnb
p1 = pj['room_type'].value_counts()
```

In [23]:
```
## Dsiplaying the output
p1
```

Out[23]:
```
room_type
Entire home/apt    51987
Private room       44887
Shared room         2149
Hotel room           115
Name: count, dtype: int64
```

In [24]:
```
## Which room type adheres to more strict cancellation policy
p2 = pj[pj['cancellation_policy']== 'strict']
```

In [25]:
```
## Displaying the output
p2
```

Out[25]:

	name	host_identity_verified	host_name	neighbourhood_group	neighbourhood	lat	
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.9
8	Large Furnished Room Near B'way	verified	Evelyn	Manhattan	Hell's Kitchen	40.76489	-73.9
9	Cozy Clean Guest Room - Family Apt	unconfirmed	Carl	Manhattan	Upper West Side	40.80178	-73.9
12	Central Manhattan/near Broadway	verified	Michael	Manhattan	Hell's Kitchen	40.76076	-73.9
24	CBG Helps Haiti Rm #2	unconfirmed	Charlotte	Brooklyn	Park Slope	40.68001	-73.9
...	...	...	...	...	...	...	
102037	Bx Apartment	unconfirmed	Vii	Bronx	Olinville	40.88438	-73.8
102040	Room in Queens, NY, near LGA.	verified	Sonia	Queens	East Elmhurst	40.76245	-73.8
102042	Central Park Views - Private Room & Bathroom	verified	Michael	Manhattan	Upper West Side	40.79712	-73.9
102049	MASTER Cozy Bedroom Queen size 2 blocks Timesq...	verified	Michael	Manhattan	Hell's Kitchen	40.76125	-73.9
102056	Room & private bathroom in historic Harlem	unconfirmed	Shireen	Manhattan	Harlem	40.81248	-73.9

32926 rows × 22 columns

In [26]:
```
## List the prices by neighborhood group and also mention which is the most expensive ne
p3 = pj['price'].groupby(pj['neighbourhood_group']).mean().sort_values(ascending=False)
p4 = pj['price'].groupby(pj['neighbourhood_group']).mean().sort_values(ascending=True)
```

In [27]:
```
## Display the output of P3
p3
```

Out[27]:
```
neighbourhood_group
Queens           628.668822
Brooklyn         625.471627
Bronx            625.271511
Staten Island    625.060870
Manhattan        621.666140
brookln          580.000000
manhatan         460.000000
Name: price, dtype: float64
```

In [28]:
```
## Display the output of p4
p4
```

Out[28]:
```
neighbourhood_group
manhatan         460.000000
brookln          580.000000
Manhattan        621.666140
Staten Island    625.060870
Bronx            625.271511
Brooklyn         625.471627
Queens           628.668822
Name: price, dtype: float64
```

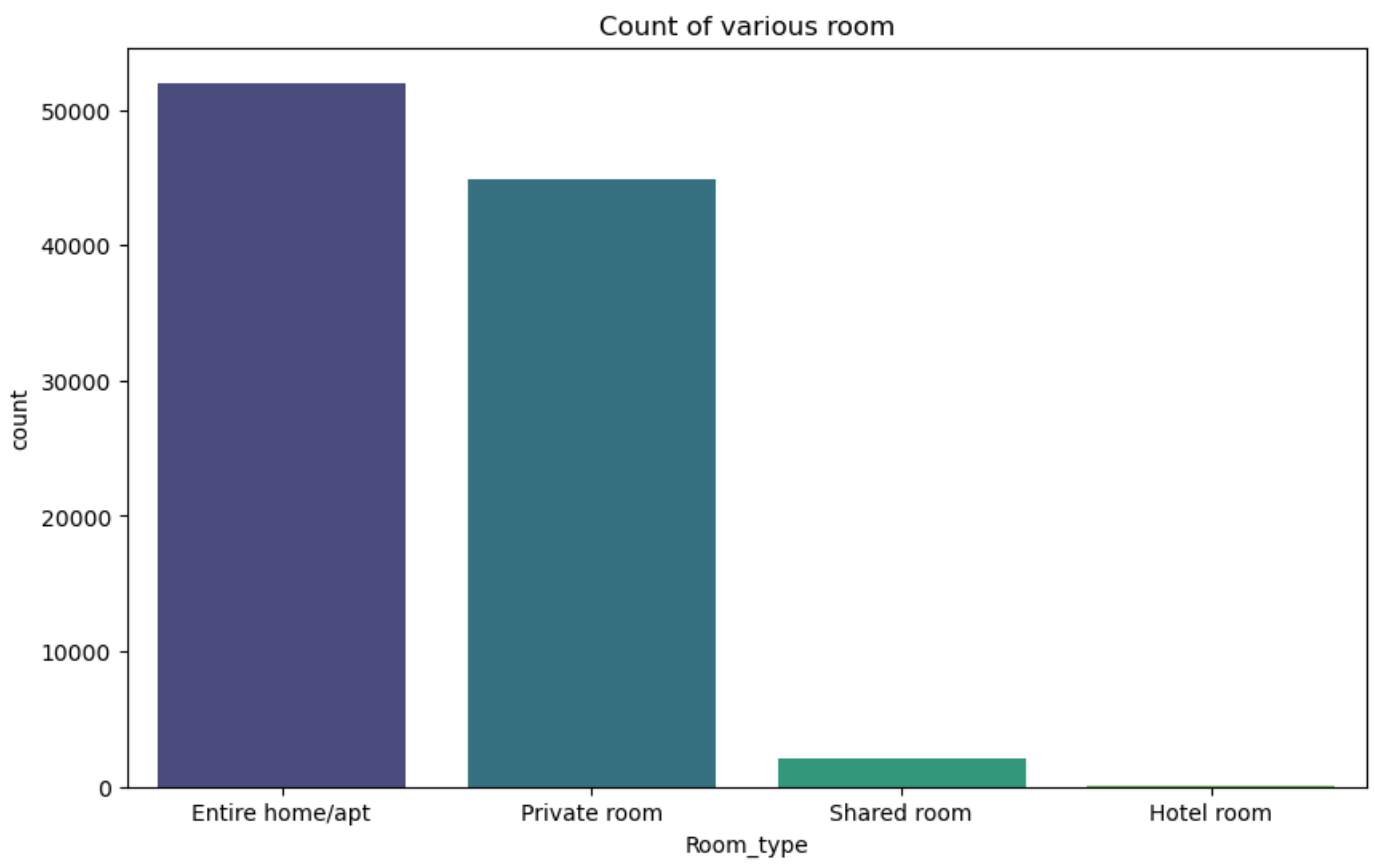
In [ ]:
```
## the most expensive room to rent from
Queens neighbourhood is the most expensive
```

Task 5a: Data Visualization (Any Tool)

- List the count of various room types avaliable with Airnb
- Which room type adheres to more strict cancellation policy
- List the prices by neighborhood group and also mention which is the most expensive neighborhood group for rentals
- List the top 10 neighborhoods in the increasing order of their price with the help of a horizontal bar graph. Which is the cheapest neighborhood.
- List the neighborhoods which offer short term rentals within 10 days. Illustrate with a bar graph
- List the prices with respect to room type using a bar graph and also state your inferences.
- Create a pie chart that shows distribution of booked days for each neighborhood group .Which neighborhood has the highest booking percentage.

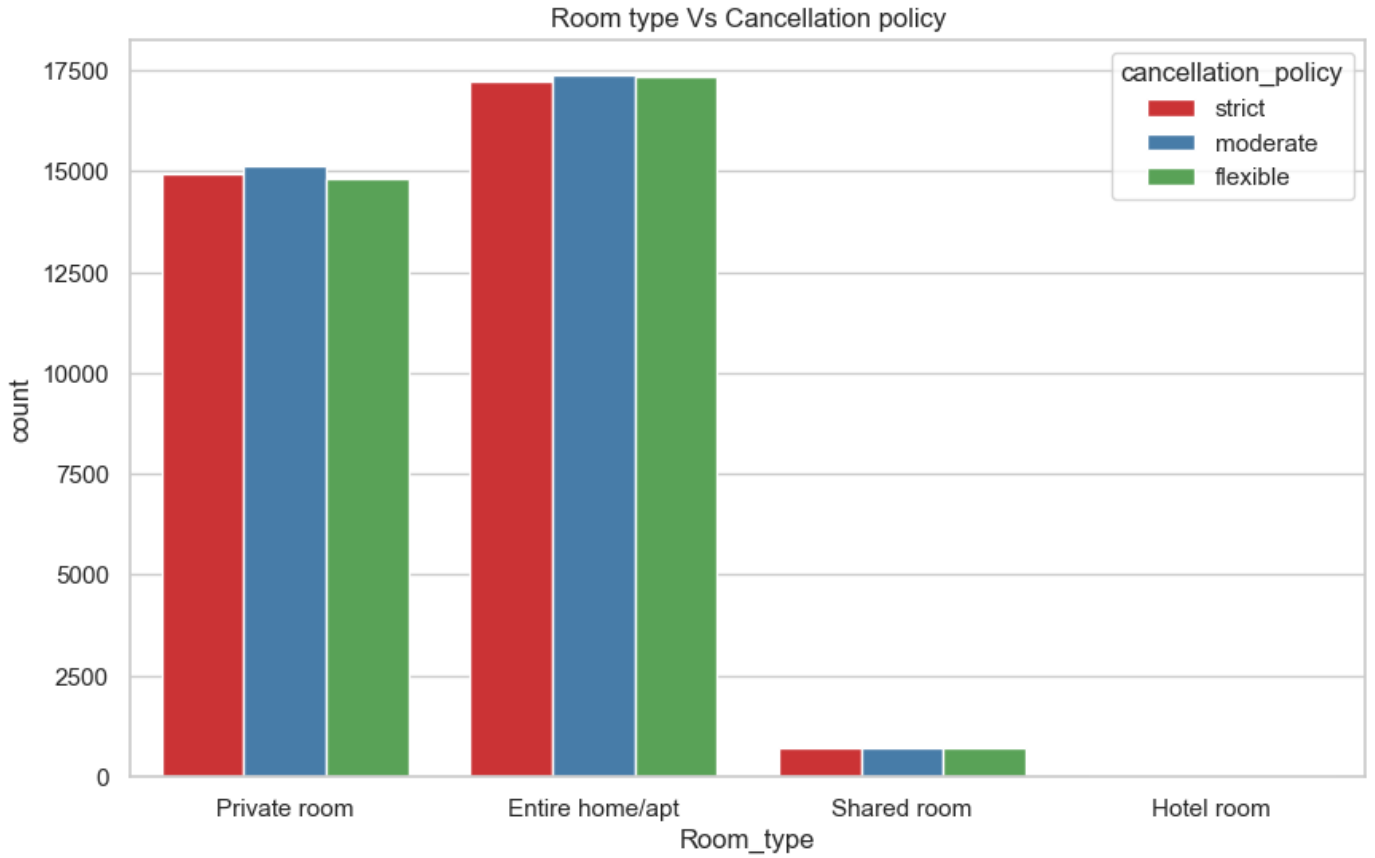
If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [29]:
```
## Count Of Various Rooms
plt.figure(figsize=(10,6))
sns.barplot(x=p1.index,y=p1.values, palette="viridis")
plt.title('Count of various room')
plt.xlabel('Room_type')
plt.ylabel('count')
plt.show()
```

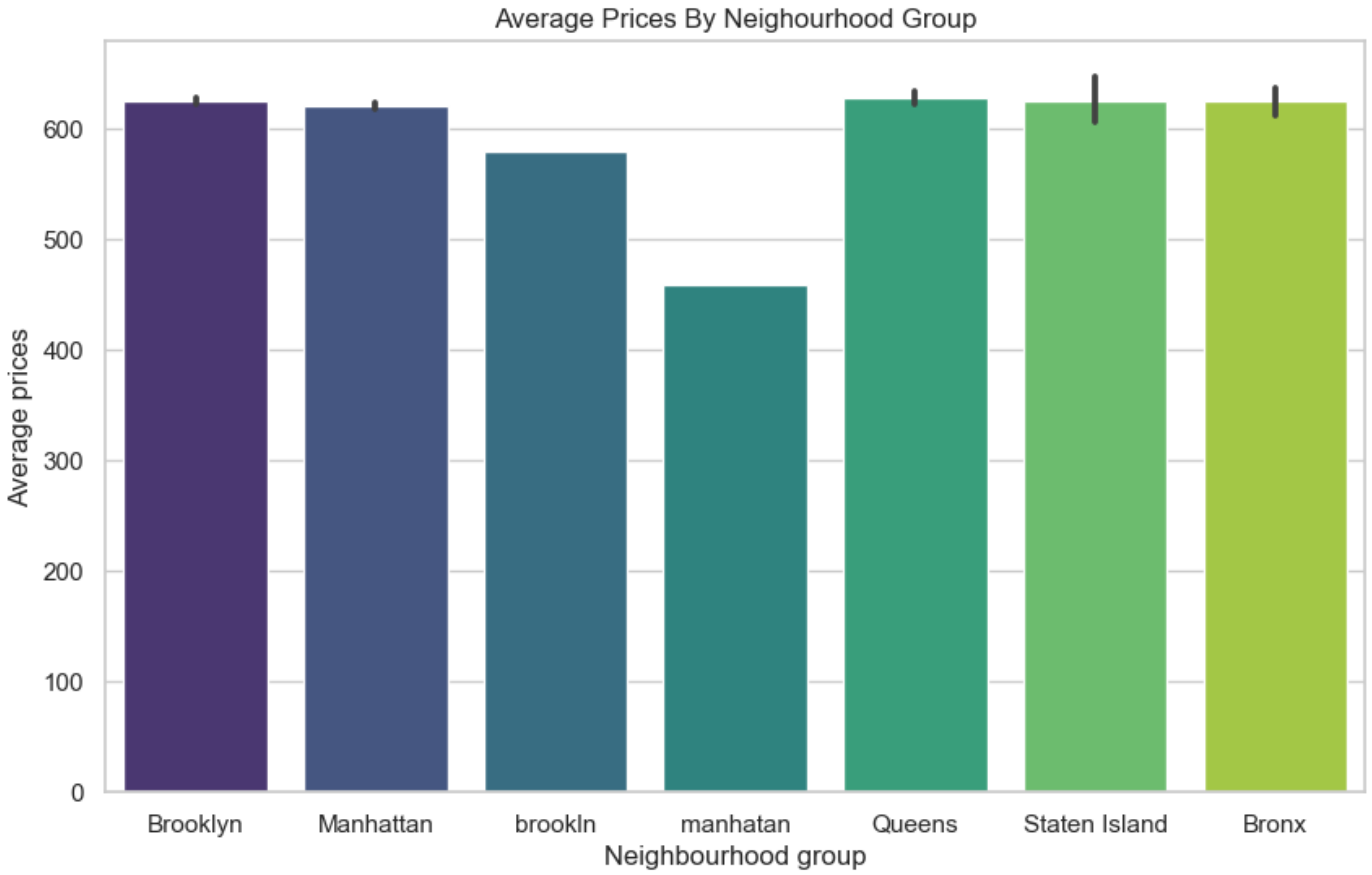


In [30]:

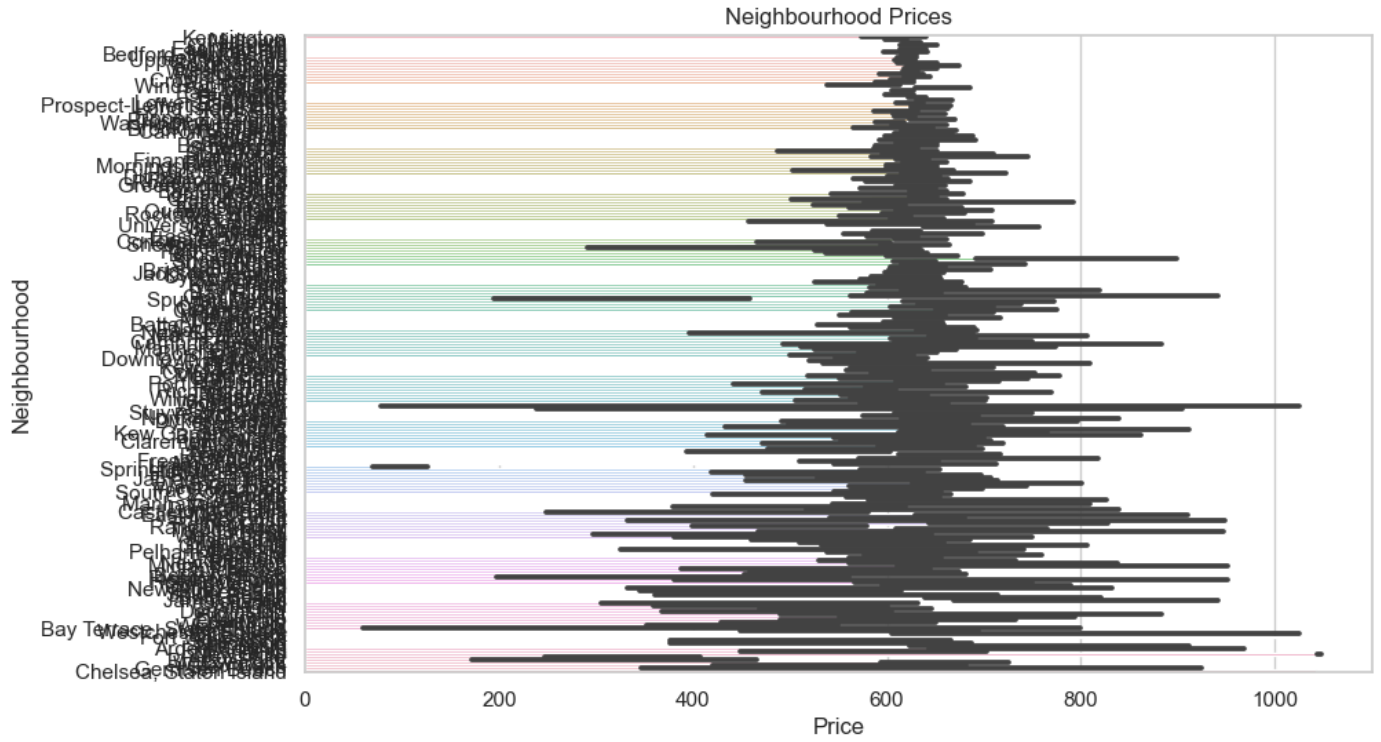
```
## Which room type adheres to more strict cancellation policy
sns.set(style="whitegrid")
plt.figure(figsize=(10,6))
sns.countplot(x='room_type', hue='cancellation_policy', data=pj,palette='Set1')
plt.title('Room type Vs Cancellation policy')
plt.xlabel('Room_type')
plt.ylabel('count')
plt.show()
```



In [31]:
```
## List the prices by neighborhood group and also mention which is the most expensive ne
plt.figure(figsize=(10,6))
sns.barplot(x='neighbourhood_group', y='price', data=pj,palette="viridis")
plt.title('Average Prices By Neighourhood Group')
plt.xlabel('Neighbourhood group')
plt.ylabel('Average prices')
plt.show()
```



In [32]:
```
## List the top 10 neighborhoods in the increasing order of their price with the help of
plt.figure(figsize=(10, 6))
sns.barplot(x='price', y='neighbourhood', data=pj, orient='h')
plt.title('Neighbourhood Prices')
plt.xlabel('Price')
plt.ylabel('Neighbourhood')
plt.show()
```



In [33]:
```
## List the neighborhoods which offer short term rentals within 10 days. Illustrate with
p5 = pj[pj['minimum_nights']<=10]
```

In [34]:
```
## Displaying the short term rentals
p5.head()
```

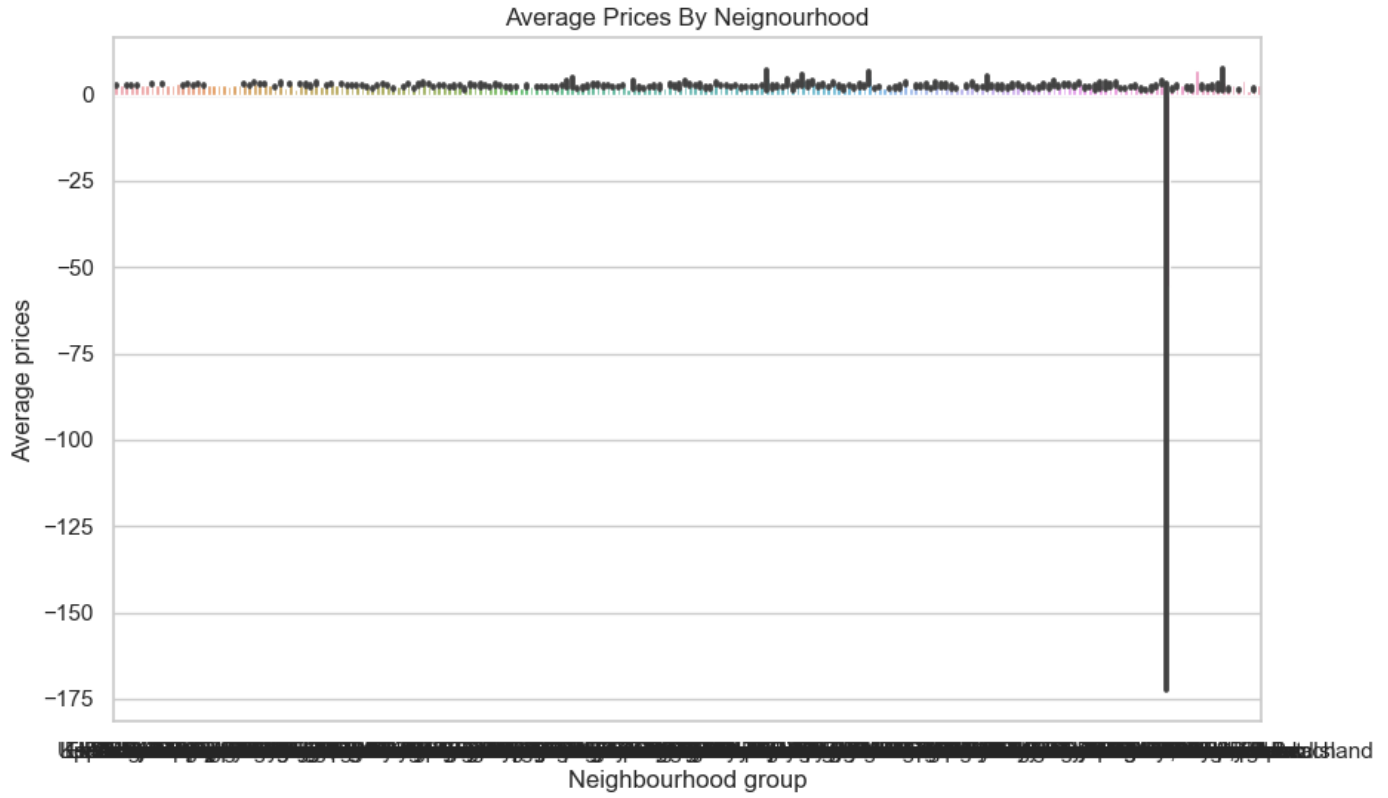
Out[34]:

	name	host_identity_verified	host_name	neighbourhood_group	neighbourhood	lat	long
0	Clean & quiet apt home by the park	unconfirmed	Madaline	Brooklyn	Kensington	40.64749	-73.97237
2	THE VILLAGE OF HARLEM....NEW YORK !	unconfirmed	Elise	Manhattan	Harlem	40.80902	-73.94190
4	Entire Apt: Spacious Studio/Loft by central park	verified	Lyndon	Manhattan	East Harlem	40.79851	-73.94399
5	Large Cozy 1 BR Apartment In Midtown East	verified	Michelle	Manhattan	Murray Hill	40.74767	-73.97500
8	Large Furnished Room Near B'way	verified	Evelyn	Manhattan	Hell's Kitchen	40.76489	-73.98493

5 rows × 22 columns

In [35]:
```
## displaying the graphs
plt.figure(figsize=(10, 6))
sns.barplot(x='neighbourhood', y='minimum_nights', data=p5)
plt.title('Average Prices By Neignourhood')
eighbourhood group')
```

```
plt.ylabel('Average prices')
plt.show()
```

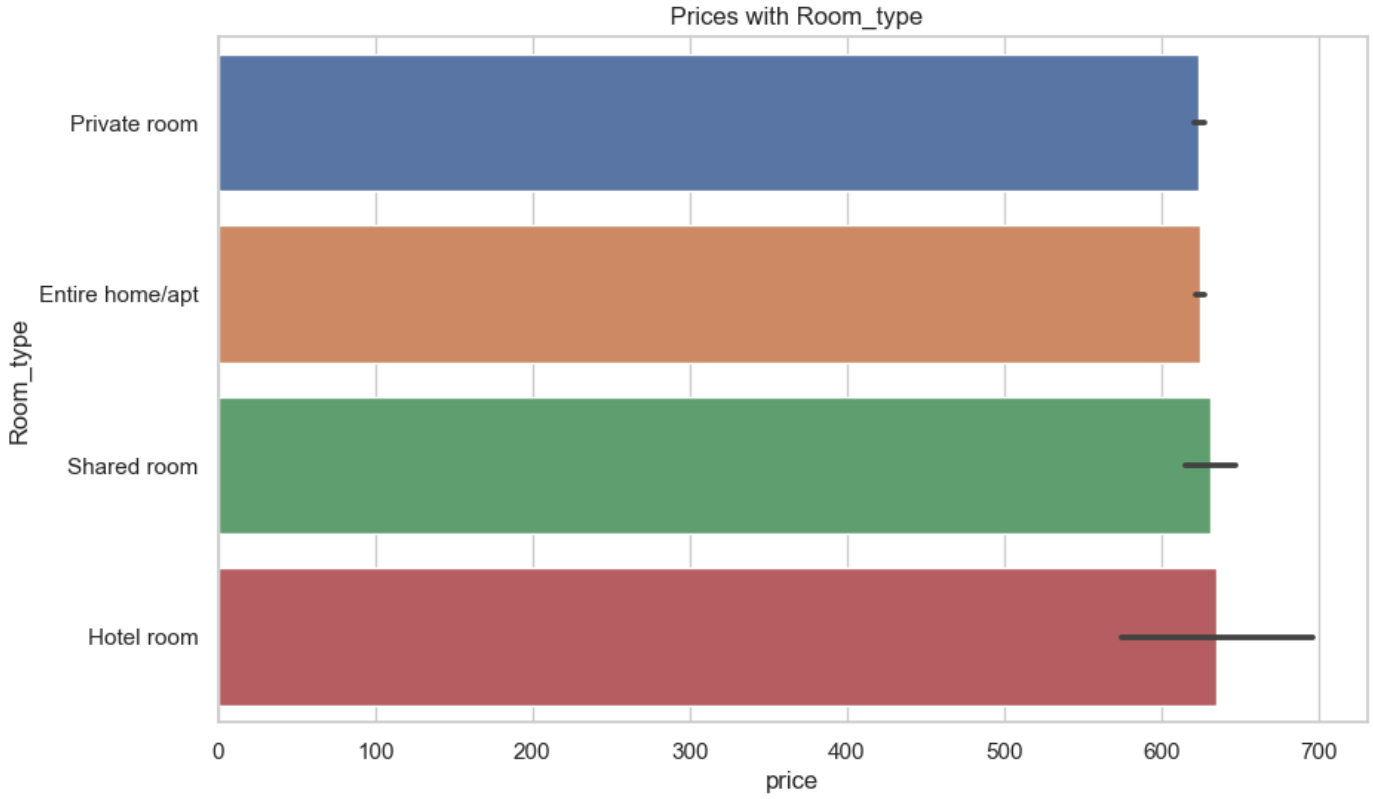


In [36]:
```
## List the prices with respect to room type using a bar graph and also state your infer
p6 = pj['price'].groupby(pj['room_type']).mean()
```

In [37]:
```
## Display the result
p6
```

Out[37]:
```
room_type
Entire home/apt    624.227711
Hotel room         666.391304
Private room       623.842516
Shared room        630.912517
Name: price, dtype: float64
```

In [38]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x='price', y='room_type', data=p5)
plt.title(' Prices with Room_type')
plt.xlabel('price')
plt.ylabel('Room_type')
plt.show()
```

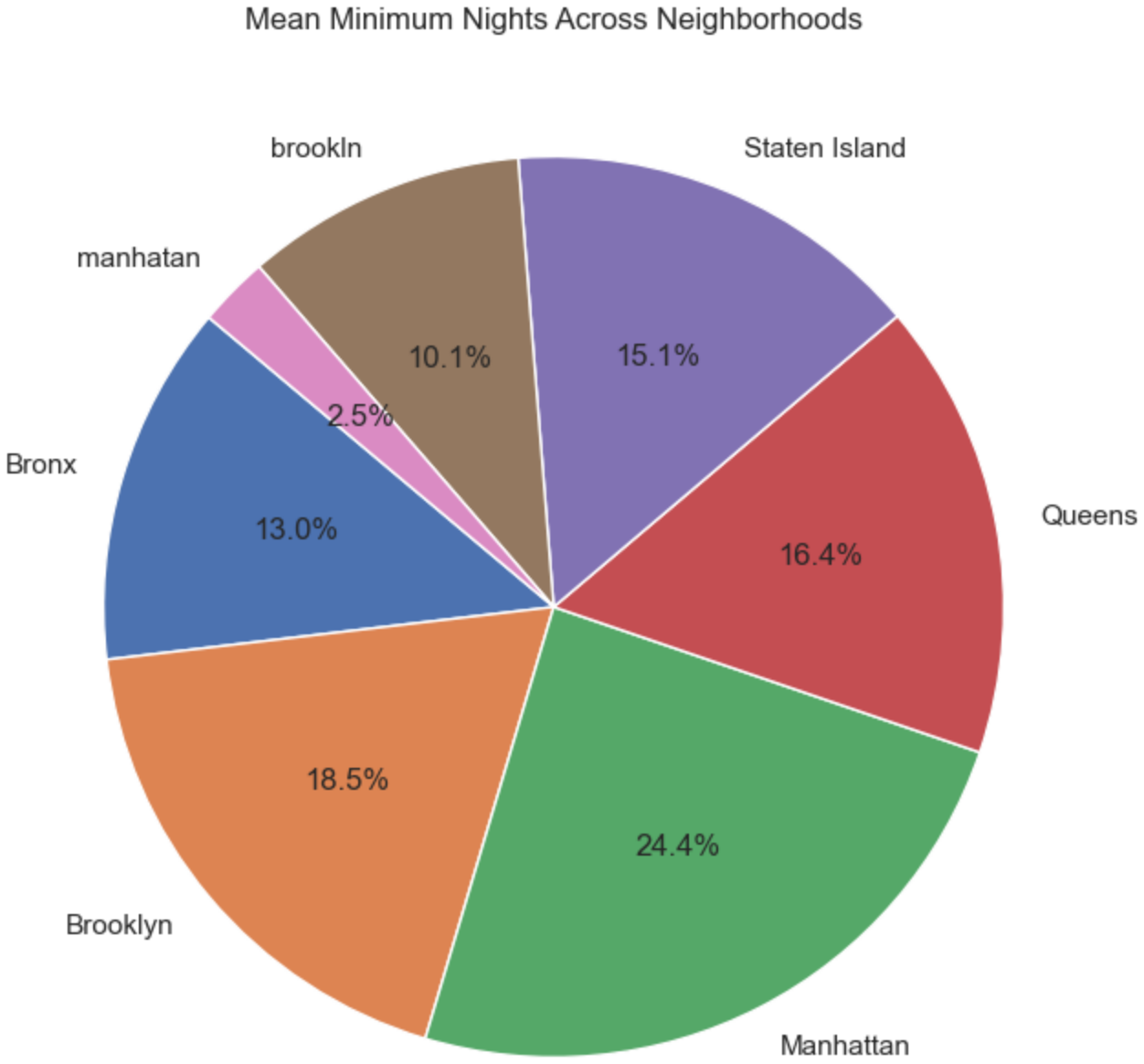


In [39]:
```
## Create a pie chart that shows distribution of booked days for each neighborhood group
p7 = pj['minimum_nights'].groupby(pj['neighbourhood_group']).mean()
```

In [40]:
```
## Output the result
p7
```

Out[40]:
```
neighbourhood_group
Bronx            5.138050
Brooklyn         7.335918
Manhattan        9.650075
Queens           6.496650
Staten Island    5.972826
brookln          4.000000
manhatan         1.000000
Name: minimum_nights, dtype: float64
```

In [42]:
```
plt.figure(figsize=(8, 8))
p7.plot(kind='pie', autopct='%1.1f%%', startangle=140)
plt.title('Mean Minimum Nights Across Neighborhoods')
plt.ylabel('')  # Removing the y-label for cleaner visualization
plt.show()
```

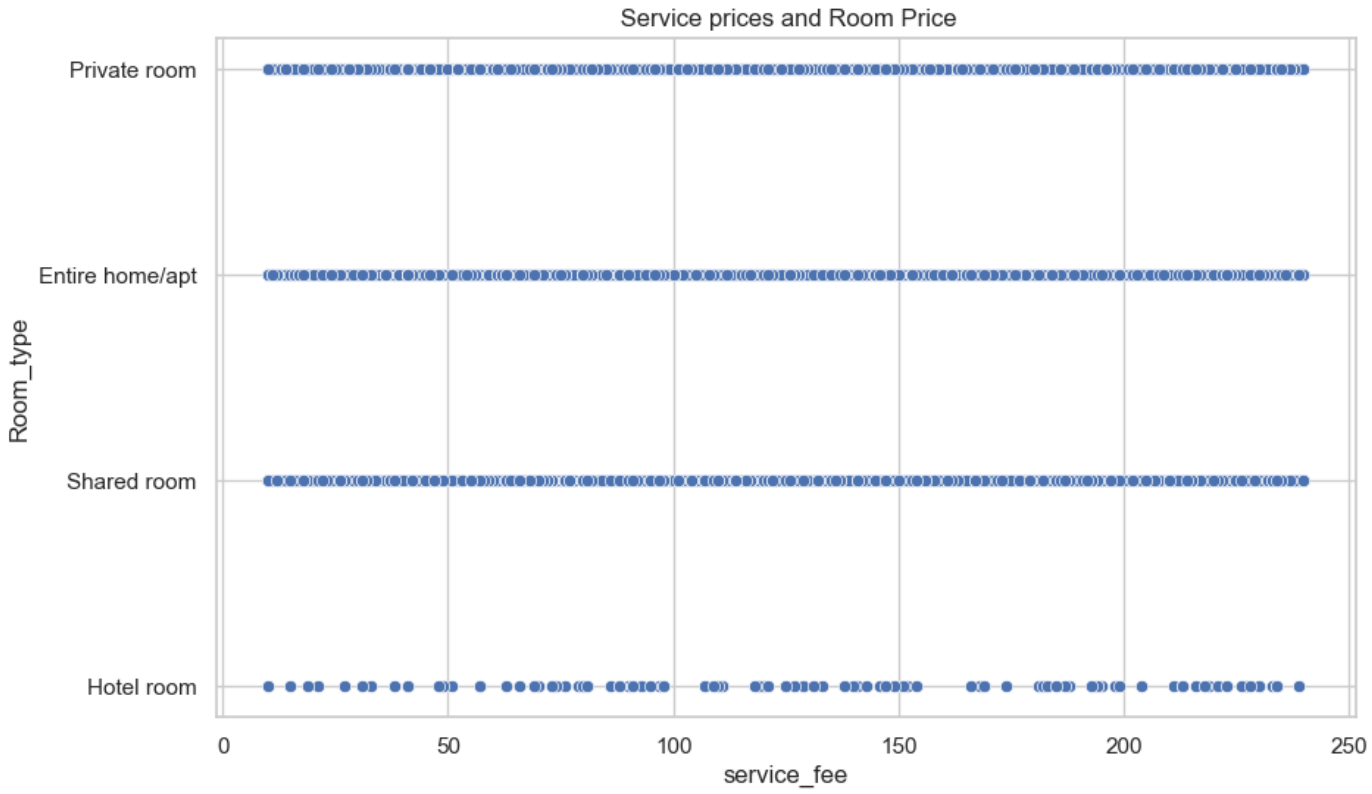


Task 5b: Data Visualization (Any Tool)

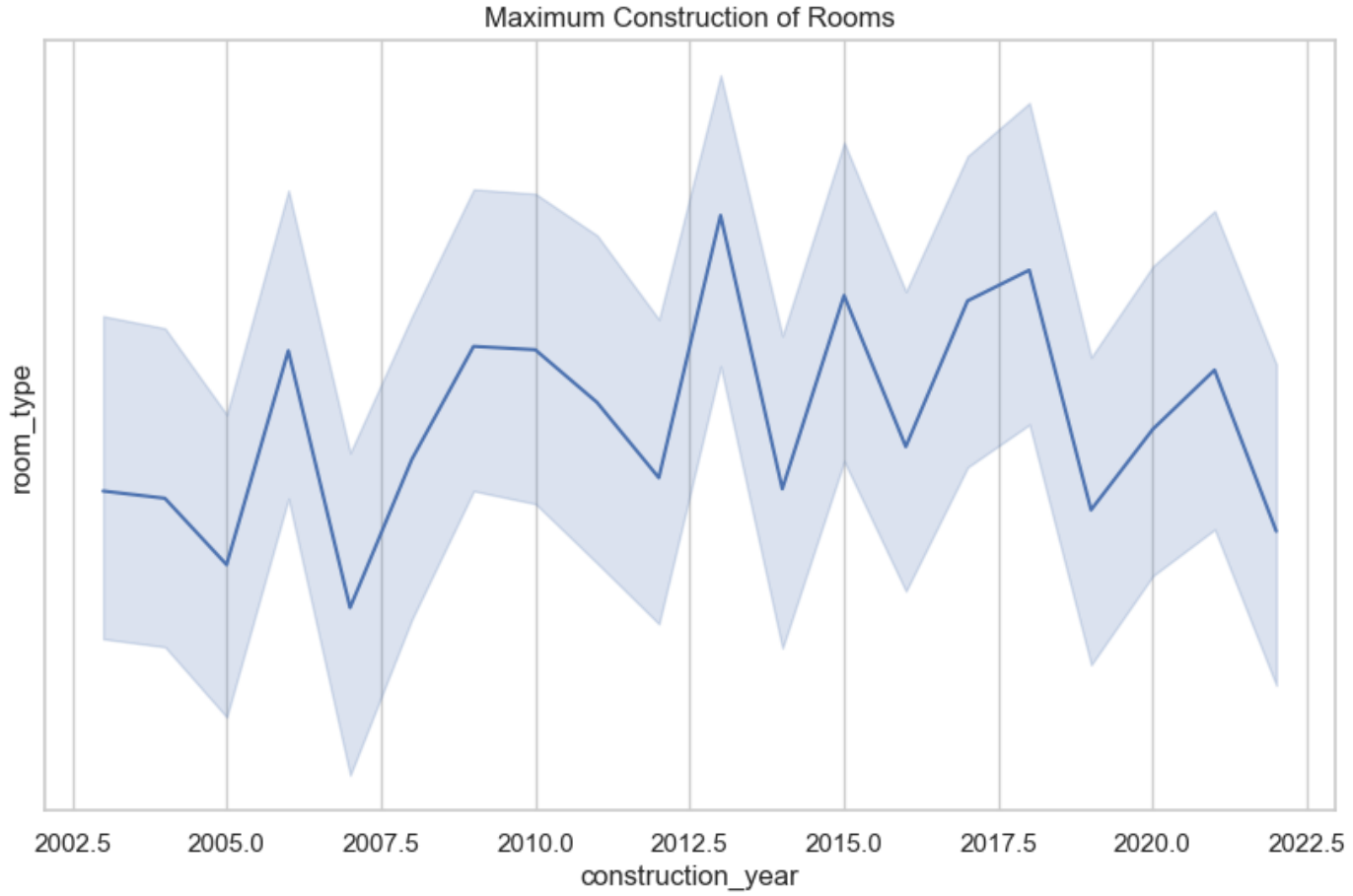
- Does service price and room price have an impact on each other. Illustrate this relationship with a scatter plot and state your inferences
- Using a line graph show in which year the maximum construction of rooms took place.

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [43]:
```
## Does service price and room price have an impact on each other. Illustrate this relat
plt.figure(figsize=(10, 6))
sns.scatterplot(x="service_fee",y="room_type",data = pj)
plt.title(' Service prices and Room Price')
plt.xlabel('service_fee')
plt.ylabel('Room_type')
plt.show()
```



In [44]:
```
## Using a line graph show in which year the maximum construction of rooms took place.
plt.figure(figsize=(10, 6))
sns.lineplot(x='construction_year',y='room_type',data=pj)
plt.title('Maximum Construction of Rooms')
plt.xlabel('construction_year')
plt.ylabel('room_type')
plt.show()
```

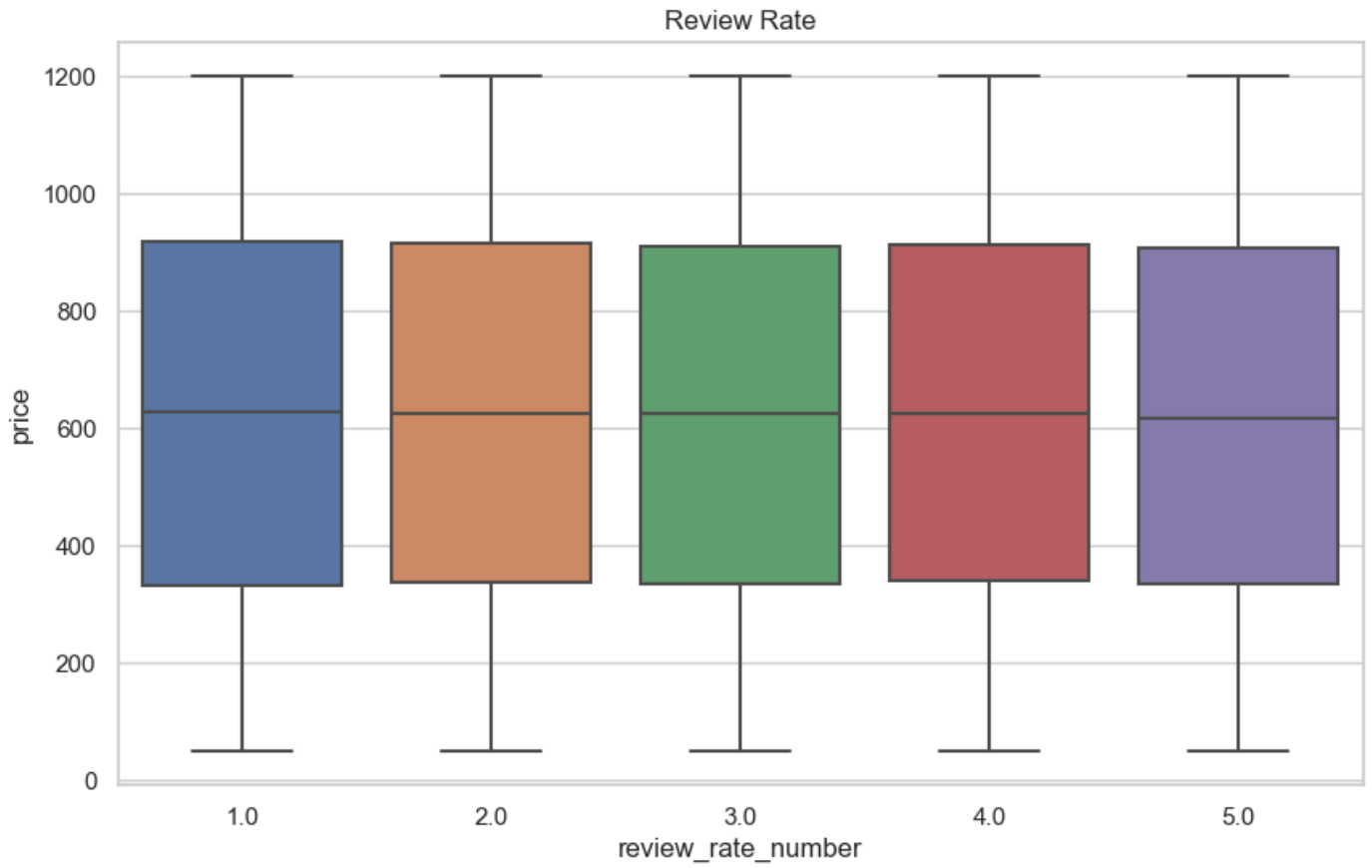


Task 5c: Data Visualization (Any Tool)

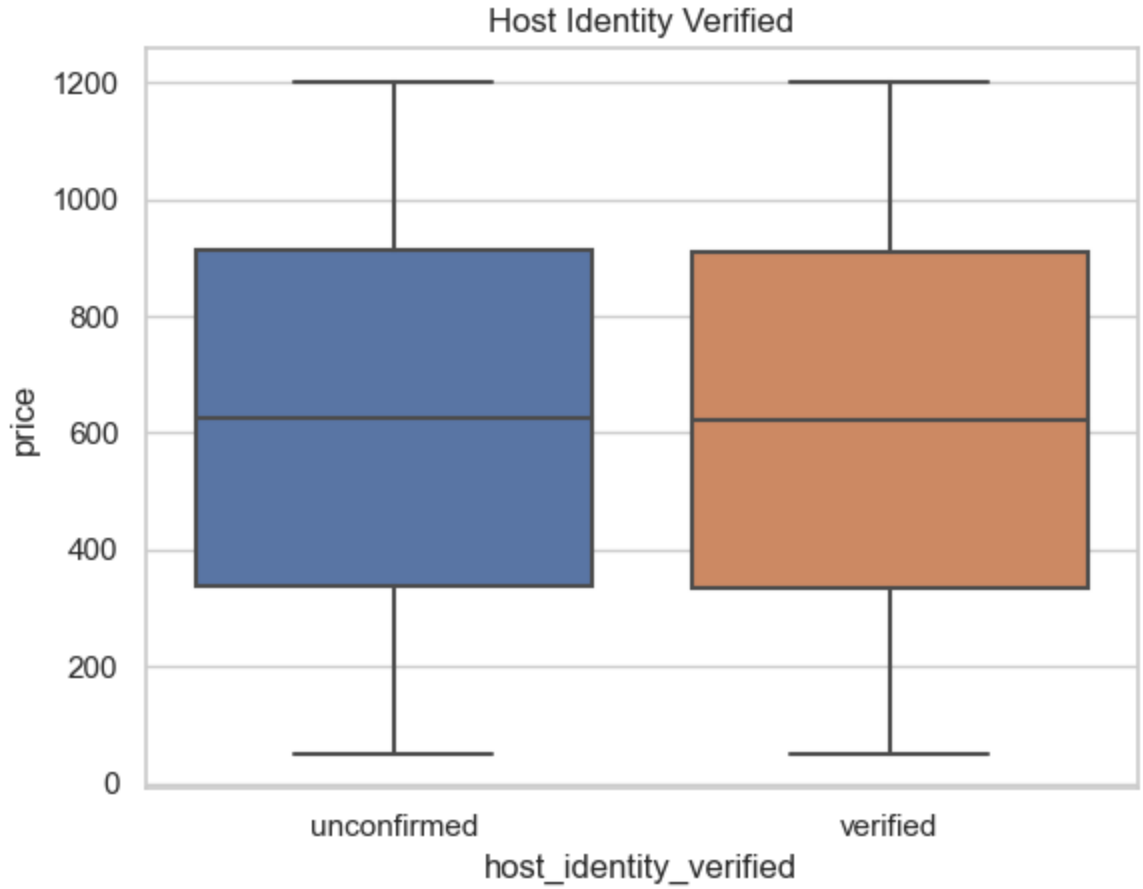
- With the help of box plots illustrate the following
  - Effect of Review Rate number on price
  - Effect of host identity verified on price

If using Python for this exercise, please include the code in the cells below. If using any other tool, please include screenshoots of your work.

In [45]:
```
#effect of review rate number on price
plt.figure(figsize=(10, 6))
sns.boxplot(x='review_rate_number',y='price',data=pj)
plt.title('Review Rate')
plt.xlabel('review_rate_number')
plt.ylabel('price')
plt.show()
```



In [46]:
```
#Effect of host identity verified on price
sns.boxplot(x='host_identity_verified',y='price',data=pj)
plt.title('Host Identity Verified')
plt.xlabel('host_identity_verified')
plt.ylabel('price')
plt.show()
```



In [ ]: