In [218…
```
greeting = "Assalam-o-Alaikum!"
print(greeting)
```

```
Assalam-o-Alaikum!
```

# Import Libraries

In [219…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import calendar
```

# Import Dataset

In [220…
```
df = pd.read_csv("AB_NYC_2019.csv")
df.head(5)
```

Out[220]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_nigh |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | |
| **1** | 2595 | Skylit Midtown Castle | 2845 | Jennifer | Manhattan | Midtown | 40.75362 | -73.98377 | Entire home/apt | 225 | |
| **2** | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Elisabeth | Manhattan | Harlem | 40.80902 | -73.94190 | Private room | 150 | |
| **3** | 3831 | Cozy Entire Floor of Brownstone | 4869 | LisaRoxanne | Brooklyn | Clinton Hill | 40.68514 | -73.95976 | Entire home/apt | 89 | |
| **4** | 5022 | Entire Apt: Spacious Studio/Loft by central park | 7192 | Laura | Manhattan | East Harlem | 40.79851 | -73.94399 | Entire home/apt | 80 | |

In [221…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   name                            48879 non-null  object 
 2   host_id                         48895 non-null  int64  
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

In [222…
```
df.describe()
```

Out[222]:

| | id | host_id | latitude | longitude | price | minimum_nights | number_of_reviews | reviews_per_month | cal |
|---|---|---|---|---|---|---|---|---|---|
| **count** | 4.889500e+04 | 4.889500e+04 | 48895.000000 | 48895.000000 | 48895.000000 | 48895.000000 | 48895.000000 | 38843.000000 | |
| **mean** | 1.901714e+07 | 6.762001e+07 | 40.728949 | -73.952170 | 152.720687 | 7.029962 | 23.274466 | 1.373221 | |
| **std** | 1.098311e+07 | 7.861097e+07 | 0.054530 | 0.046157 | 240.154170 | 20.510550 | 44.550582 | 1.680442 | |
| **min** | 2.539000e+03 | 2.438000e+03 | 40.499790 | -74.244420 | 0.000000 | 1.000000 | 0.000000 | 0.010000 | |
| **25%** | 9.471945e+06 | 7.822033e+06 | 40.690100 | -73.983070 | 69.000000 | 1.000000 | 1.000000 | 0.190000 | |
| **50%** | 1.967728e+07 | 3.079382e+07 | 40.723070 | -73.955680 | 106.000000 | 3.000000 | 5.000000 | 0.720000 | |
| **75%** | 2.915218e+07 | 1.074344e+08 | 40.763115 | -73.936275 | 175.000000 | 5.000000 | 24.000000 | 2.020000 | |
| **max** | 3.648724e+07 | 2.743213e+08 | 40.913060 | -73.712990 | 10000.000000 | 1250.000000 | 629.000000 | 58.500000 | |

# Data Analysis

**1. What is the average price of the rooms?**

In [223…
```
average_price = df["price"].agg("mean").round(2)
print("Average Price Of the Room = $" + str(average_price))
```

```
Average Price Of the Room = $152.72
```

**2. What is the minimum and maximum number of nights for the bookings?**

In [224…
```
maximum = df["minimum_nights"].max()
print("Maximum Number oF Nights for Booking = " + str(maximum))
```

```
Maximum Number oF Nights for Booking = 1250
```

In [225…
```
minimum = df["minimum_nights"].min()
print("Minimum Number Of Nights For Booking = " + str(minimum))
```

```
Minimum Number Of Nights For Booking = 1
```

**3. How many reviews does each room type have on average?**

In [226…
```
reviews = df.groupby("room_type")["number_of_reviews"].agg("mean").to_frame().reset_index().round(2).sort_value
reviews.columns = ["Room Type", "Average Reviews"]
reviews
```

Out[226]:

| | Room Type | Average Reviews |
|---|---|---|
| 1 | Private room | 24.11 |
| 0 | Entire home/apt | 22.84 |
| 2 | Shared room | 16.60 |

**4. What is the most recent review date recorded?**

In [227…
```
recent = df["last_review"].dropna()
recent = pd.to_datetime(recent).to_frame().sort_values("last_review", ascending = False)
recent = recent.iloc[0]
print("The most recent review date recorded is = " + str(recent["last_review"]))
```

```
The most recent review date recorded is = 2019-07-08 00:00:00
```

In [228…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48895 entries, 0 to 48894
Data columns (total 16 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   id                              48895 non-null  int64  
 1   name                            48879 non-null  object 
 2   host_id                         48895 non-null  int64  
 3   host_name                       48874 non-null  object 
 4   neighbourhood_group             48895 non-null  object 
 5   neighbourhood                   48895 non-null  object 
 6   latitude                        48895 non-null  float64
 7   longitude                       48895 non-null  float64
 8   room_type                       48895 non-null  object 
 9   price                           48895 non-null  int64  
 10  minimum_nights                  48895 non-null  int64  
 11  number_of_reviews               48895 non-null  int64  
 12  last_review                     38843 non-null  object 
 13  reviews_per_month               38843 non-null  float64
 14  calculated_host_listings_count  48895 non-null  int64  
 15  availability_365                48895 non-null  int64  
dtypes: float64(3), int64(7), object(6)
memory usage: 6.0+ MB
```

**5. What is the average number of reviews per month?**

In [229…
```
average_review = df["reviews_per_month"].agg("mean").round(2)
print("Average reviews per Month = " + str(average_review))
```

```
Average reviews per Month = 1.37
```

**6. How many unique hosts are there in the dataset?**

In [230…
```
uni_hosts = df["host_id"].nunique()
print("Total number of Unique Hosts in Dataset are = " + str(uni_hosts))
```

```
Total number of Unique Hosts in Dataset are = 37457
```

**7. What is the average availability of the rooms in terms of the number of days in a year?**

In [231…
```
rooms = df["availability_365"].agg("mean").round(2)
print("The Average Availability of the rooms in terms of the number of Days in a Year = " + str(rooms))
```

```
The Average Availability of the rooms in terms of the number of Days in a Year = 112.78
```

**8. What is the most common room type?**

In [232…
```
common_room = df["room_type"].value_counts().to_frame().reset_index().sort_values("room_type", ascending = Fals
common_room.columns =["Room Type", "Counts"]
common_room = common_room.iloc[0]
print("Common Room Type", "'"+common_room["Room Type"]+"'", "has most common rooms which are = " + str(common_r
```

```
Common Room Type 'Entire home/apt' has most common rooms which are = 25409
```

In [233…
```
df.head(1)
```

Out[233]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_nights | number |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | 1 | |

**9. What is the range of prices for each room type?**

In [234…
```
range_price = df.groupby("room_type")["price"].agg(["max", "min"]).reset_index()
range_price
```

Out[234]:

| | room_type | max | min |
|---|---|---|---|
| **0** | Entire home/apt | 10000 | 0 |
| **1** | Private room | 10000 | 0 |
| **2** | Shared room | 1800 | 0 |

**10. How many rooms have zero reviews?**

In [235…
```
zero_reviews = df[df["number_of_reviews"] == 0]
zero_reviews
```

Out[235]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | mini |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **2** | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Elisabeth | Manhattan | Harlem | 40.80902 | -73.94190 | Private room | 150 | |
| **19** | 7750 | Huge 2 BR Upper East Cental Park | 17985 | Sing | Manhattan | East Harlem | 40.79685 | -73.94872 | Entire home/apt | 190 | |
| **26** | 8700 | Magnifique Suite au N de Manhattan - vue Cloitres | 26394 | Claude & Sophie | Manhattan | Inwood | 40.86754 | -73.92639 | Private room | 80 | |
| **36** | 11452 | Clean and Quiet in Brooklyn | 7355 | Vt | Brooklyn | Bedford-Stuyvesant | 40.68876 | -73.94312 | Private room | 35 | |
| **38** | 11943 | Country space in the city | 45445 | Harriet | Brooklyn | Flatbush | 40.63702 | -73.96327 | Private room | 150 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **48890** | 36484665 | Charming one bedroom - newly renovated rowhouse | 8232441 | Sabrina | Brooklyn | Bedford-Stuyvesant | 40.67853 | -73.94995 | Private room | 70 | |
| **48891** | 36485057 | Affordable room in Bushwick/East Williamsburg | 6570630 | Marisol | Brooklyn | Bushwick | 40.70184 | -73.93317 | Private room | 40 | |
| **48892** | 36485431 | Sunny Studio at Historical Neighborhood | 23492952 | Ilgar & Aysel | Manhattan | Harlem | 40.81475 | -73.94867 | Entire home/apt | 115 | |
| **48893** | 36485609 | 43rd St. Time Square-cozy single bed | 30985759 | Taz | Manhattan | Hell's Kitchen | 40.75751 | -73.99112 | Shared room | 55 | |
| **48894** | 36487245 | Trendy duplex in the very heart of Hell's Kitchen | 68119814 | Christophe | Manhattan | Hell's Kitchen | 40.76404 | -73.98933 | Private room | 90 | |

10052 rows × 16 columns

In [236…
```
room_count = zero_reviews["id"].nunique()
print("The Total Rooms which have zero reviews = " + str(room_count))
```

```
The Total Rooms which have zero reviews = 10052
```

**11. What is the average number of reviews for rooms with a price greater than $100?**

In [237…
```
greater_price = df[df["price"] > 100]
greater_price
```

Out[237]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | min |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | |
| **1** | 2595 | Skylit Midtown Castle | 2845 | Jennifer | Manhattan | Midtown | 40.75362 | -73.98377 | Entire home/apt | 225 | |
| **2** | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Elisabeth | Manhattan | Harlem | 40.80902 | -73.94190 | Private room | 150 | |
| **5** | 5099 | Large Cozy 1 BR Apartment In Midtown East | 7322 | Chris | Manhattan | Murray Hill | 40.74767 | -73.97500 | Entire home/apt | 200 | |
| **9** | 5238 | Cute & Cozy Lower East Side 1 bdrm | 7549 | Ben | Manhattan | Chinatown | 40.71344 | -73.99037 | Entire home/apt | 150 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **48884** | 36482783 | Brooklyn Oasis in the heart of Williamsburg | 274307600 | Jonathan | Brooklyn | Williamsburg | 40.71790 | -73.96238 | Private room | 190 | |
| **48886** | 36483010 | Comfy 1 Bedroom in Midtown East | 274311461 | Scott | Manhattan | Midtown | 40.75561 | -73.96723 | Entire home/apt | 200 | |
| **48887** | 36483152 | Garden Jewel Apartment in Williamsburg New York | 208514239 | Melki | Brooklyn | Williamsburg | 40.71232 | -73.94220 | Entire home/apt | 170 | |
| **48888** | 36484087 | Spacious Room w/ Private Rooftop, Central loca... | 274321313 | Kat | Manhattan | Hell's Kitchen | 40.76392 | -73.99183 | Private room | 125 | |
| **48892** | 36485431 | Sunny Studio at Historical Neighborhood | 23492952 | Ilgar & Aysel | Manhattan | Harlem | 40.81475 | -73.94867 | Entire home/apt | 115 | |

24967 rows × 16 columns

In [238…
```
average = greater_price.groupby("room_type")["number_of_reviews"].agg("mean").to_frame().reset_index()
average.columns = ["Room Types", "Average Reviews"]
average
```

Out[238]:

| | Room Types | Average Reviews |
|---|---|---|
| **0** | Entire home/apt | 22.039021 |
| **1** | Private room | 23.523726 |
| **2** | Shared room | 8.398438 |

**12. What is the correlation between the price and the number of reviews?**

In [239…
```
CoRRelation = df[["price", "number_of_reviews"]].corr()
CoRRelation
```

Out[239]:

| | price | number_of_reviews |
|---|---|---|
| **price** | 1.000000 | -0.047954 |
| **number_of_reviews** | -0.047954 | 1.000000 |

In [240…
```
df.head(3)
```

Out[240]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_night |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | |
| 1 | 2595 | Skylit Midtown Castle | 2845 | Jennifer | Manhattan | Midtown | 40.75362 | -73.98377 | Entire home/apt | 225 | |
| 2 | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Elisabeth | Manhattan | Harlem | 40.80902 | -73.94190 | Private room | 150 | |

**13. How many rooms have a review date after January 1, 2019?**

In [241…

```
df["last_review"] = pd.to_datetime(df["last_review"])
review_date = df[df["last_review"] > "2019-01-01"]
review_date
```

Out[241]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimu |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2595 | Skylit Midtown Castle | 2845 | Jennifer | Manhattan | Midtown | 40.75362 | -73.98377 | Entire home/apt | 225 | |
| 3 | 3831 | Cozy Entire Floor of Brownstone | 4869 | LisaRoxanne | Brooklyn | Clinton Hill | 40.68514 | -73.95976 | Entire home/apt | 89 | |
| 5 | 5099 | Large Cozy 1 BR Apartment In Midtown East | 7322 | Chris | Manhattan | Murray Hill | 40.74767 | -73.97500 | Entire home/apt | 200 | |
| 7 | 5178 | Large Furnished Room Near B'way | 8967 | Shunichi | Manhattan | Hell's Kitchen | 40.76489 | -73.98493 | Private room | 79 | |
| 9 | 5238 | Cute & Cozy Lower East Side 1 bdrm | 7549 | Ben | Manhattan | Chinatown | 40.71344 | -73.99037 | Entire home/apt | 150 | |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 48782 | 36425863 | Lovely Privet Bedroom with Privet Restroom | 83554966 | Rusaa | Manhattan | Upper East Side | 40.78099 | -73.95366 | Private room | 129 | |
| 48790 | 36427429 | No.2 with queen size bed | 257683179 | H Ai | Queens | Flushing | 40.75104 | -73.81459 | Private room | 45 | |
| 48799 | 36438336 | Seas The Moment | 211644523 | Ben | Staten Island | Great Kills | 40.54179 | -74.14275 | Private room | 235 | |
| 48805 | 36442252 | 1B-1B apartment near by Metro | 273841667 | Blaine | Bronx | Mott Haven | 40.80787 | -73.92400 | Entire home/apt | 100 | |
| 48852 | 36455809 | Cozy Private Room in Bushwick, Brooklyn | 74162901 | Christine | Brooklyn | Bushwick | 40.69805 | -73.92801 | Private room | 30 | |

24811 rows × 16 columns

In [242…

```
Review_Date = review_date["id"].agg("count")
print("Total rooms have a review date after January 1, 2019 = " + str(Review_Date))
```

```
Total rooms have a review date after January 1, 2019 = 24811
```

**14. What is the average number of reviews for rooms with a minimum stay of less than or equal to 3 nights?**

In [243…

```
night_stay = df[df["minimum_nights"] <= 3]
RRNS = night_stay["number_of_reviews"].agg("mean").round(2)
print("Average number of Reviews for rooms with a Minimum Stay of less than or Equal to 3 Nights = " + str(RRNS
```

```
Average number of Reviews for rooms with a Minimum Stay of less than or Equal to 3 Nights = 28.63
```

**15. What is the average price for rooms with an availability of more than 180 days in a year?**

In [244…

```
Room_availability = df[df["availability_365"] > 180]
Room_availability.head(5)
```

Out[244]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_nigh |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | |
| 1 | 2595 | Skylit Midtown Castle | 2845 | Jennifer | Manhattan | Midtown | 40.75362 | -73.98377 | Entire home/apt | 225 | |
| 2 | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Elisabeth | Manhattan | Harlem | 40.80902 | -73.94190 | Private room | 150 | |
| 3 | 3831 | Cozy Entire Floor of Brownstone | 4869 | LisaRoxanne | Brooklyn | Clinton Hill | 40.68514 | -73.95976 | Entire home/apt | 89 | |
| 7 | 5178 | Large Furnished Room Near B'way | 8967 | Shunichi | Manhattan | Hell's Kitchen | 40.76489 | -73.98493 | Private room | 79 | |

In [245…
```
APFR = Room_availability["price"].agg("mean").round(2)
print("The Average price for rooms with an availability of more than 180 days in a year = $" + str(APFR))
```

```
The Average price for rooms with an availability of more than 180 days in a year = $178.47
```

In [246…
```
df.head(1)
```

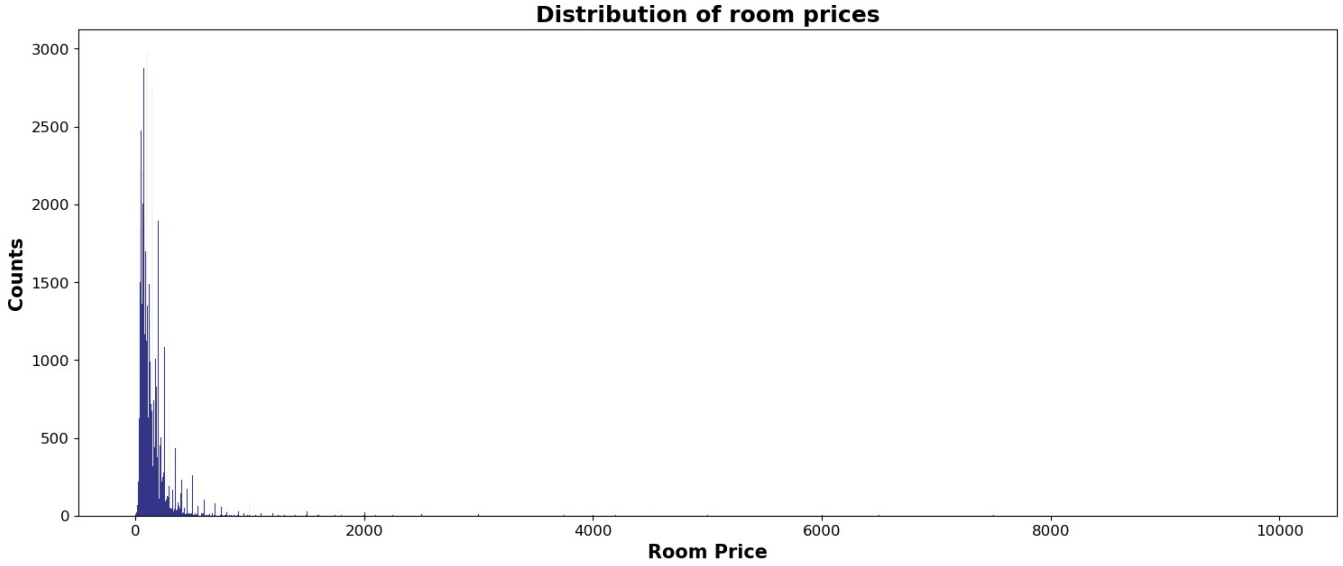
Out[246]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_nights | number |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | 1 | |

# Data Visualization

**1. Create a histogram to visualize the distribution of room prices.**

In [247…
```
plt.figure(figsize = (18, 7))
sns.histplot(x = "price", data = df, color = "navy")
plt.title("Distribution of room prices", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Room Price", weight = "bold", size = 15)
plt.ylabel("Counts", weight = "bold", size = 15)
plt.show()
```



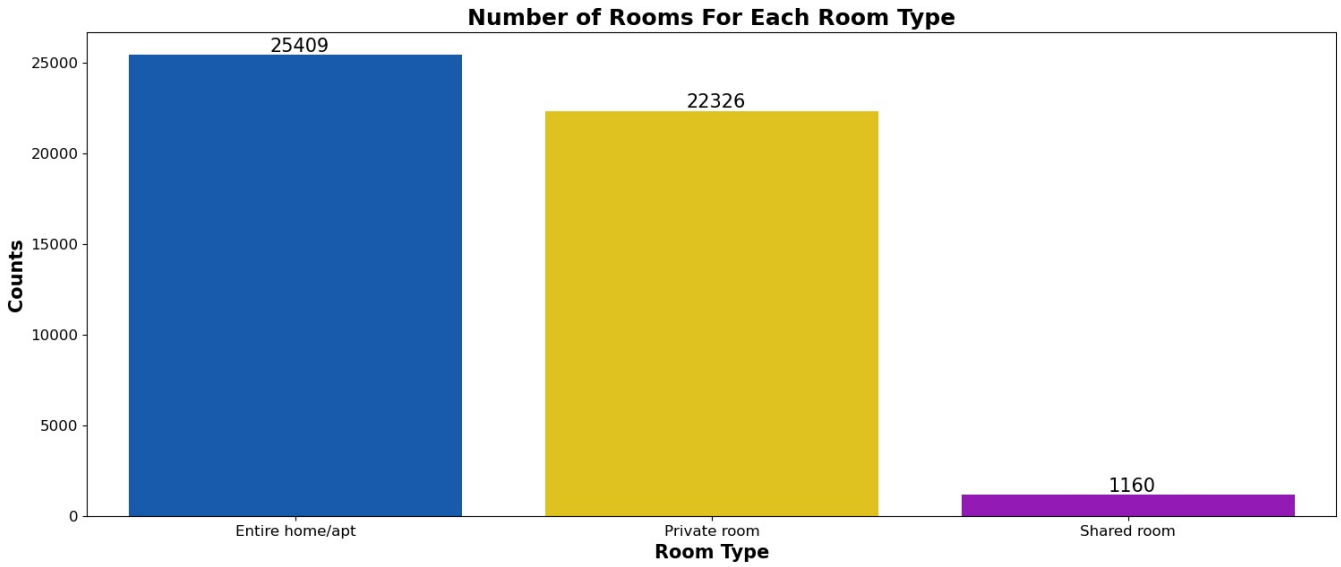

**2. Plot a bar chart to compare the number of rooms for each room type.**

In [248…
```
num_of_room = df["room_type"].value_counts().to_frame().reset_index()
num_of_room.columns = ["Room Type", "Counts"]
num_of_room
```

Out[248]:

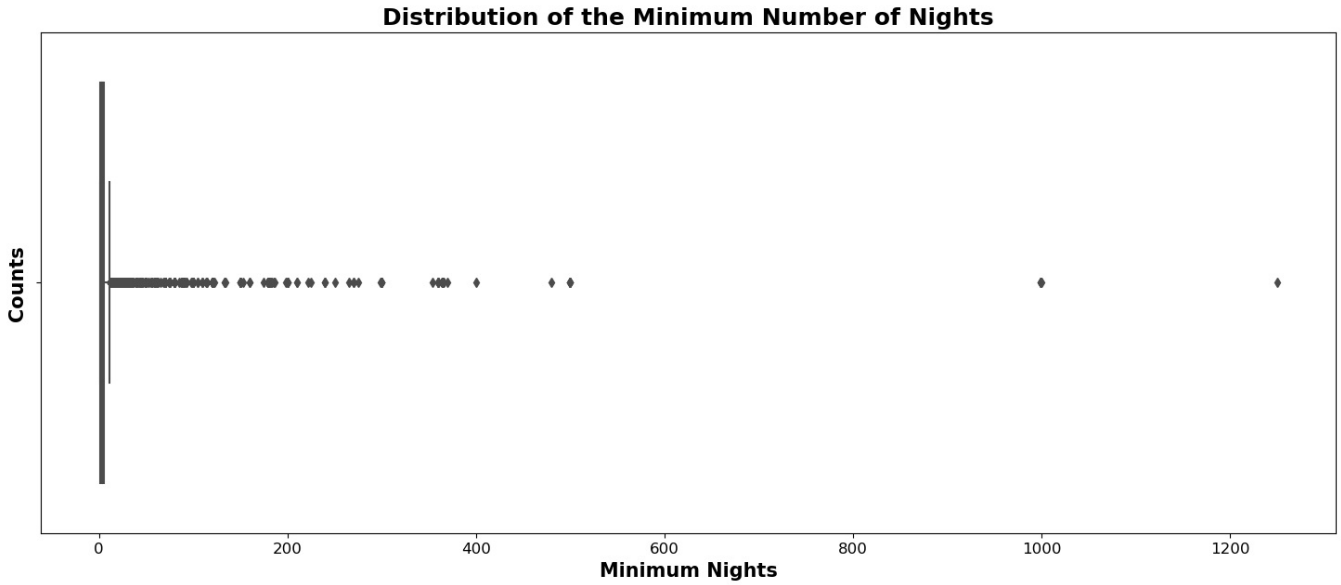
| | Room Type | Counts |
|---|---|---|
| 0 | Entire home/apt | 25409 |
| 1 | Private room | 22326 |
| 2 | Shared room | 1160 |

In [249…
```
plt.figure(figsize = (18, 7))
graph = sns.barplot(x = "Room Type", y = "Counts", data = num_of_room, palette = "prism")
for p in graph.patches:
        graph.annotate('{:.0f}'.format(p.get_height()),
                       (p.get_x()+0.41, p.get_height()),
                        ha='center', va='bottom',color= 'black', size = 15)
plt.title("Number of Rooms For Each Room Type", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Room Type", weight = "bold", size = 15)
plt.ylabel("Counts", weight = "bold", size = 15)
plt.show()
```




**3. Visualize the distribution of the minimum number of nights using a box plot.**

In [250…
```
plt.figure(figsize = (18, 7))
sns.boxplot(x = "minimum_nights", data = df, color = "cyan")
plt.title("Distribution of the Minimum Number of Nights", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Minimum Nights", weight = "bold", size = 15)
plt.ylabel("Counts", weight = "bold", size = 15)
plt.show()
```




**4. Create a line plot to show the trend of the number of reviews per month.**

In [251…
```
month_reviews = df[["last_review", "number_of_reviews"]].dropna()
month_reviews["Months"] = month_reviews["last_review"].dt.month_name()
month_reviews["Year"] = month_reviews["last_review"].dt.year
month_reviews["Month Year"] = pd.to_datetime(month_reviews['Months'] + ' ' + month_reviews['Year'].astype(str),
```

```
month_reviews = month_reviews.sort_values("last_review", ascending = True)
month_reviews
```

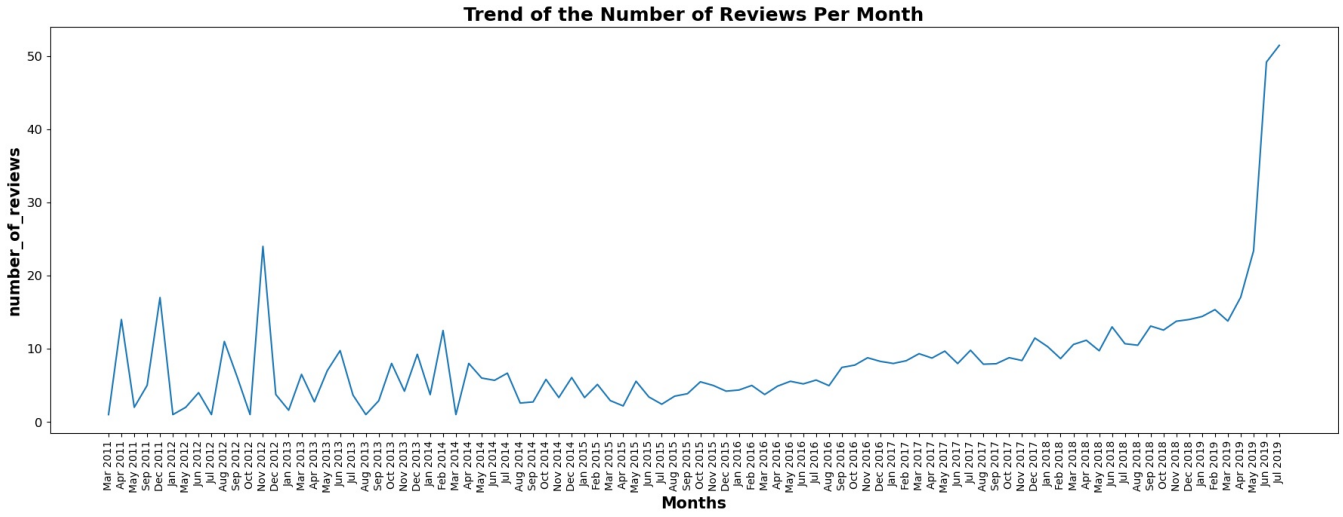
Out[251]:

| | last_review | number_of_reviews | Months | Year | Month Year |
|---|---|---|---|---|---|
| **317** | 2011-03-28 | 1 | March | 2011 | 2011-03-01 |
| **163** | 2011-04-25 | 14 | April | 2011 | 2011-04-01 |
| **330** | 2011-05-12 | 2 | May | 2011 | 2011-05-01 |
| **125** | 2011-09-18 | 9 | September | 2011 | 2011-09-01 |
| **143** | 2011-09-19 | 1 | September | 2011 | 2011-09-01 |
| **...** | ... | ... | ... | ... | ... |
| **41072** | 2019-07-08 | 19 | July | 2019 | 2019-07-01 |
| **16692** | 2019-07-08 | 193 | July | 2019 | 2019-07-01 |
| **41194** | 2019-07-08 | 40 | July | 2019 | 2019-07-01 |
| **26791** | 2019-07-08 | 89 | July | 2019 | 2019-07-01 |
| **48852** | 2019-07-08 | 1 | July | 2019 | 2019-07-01 |

38843 rows × 5 columns

In [252…

```
plt.figure(figsize = (22, 7))
sns.lineplot(x=month_reviews['Month Year'].dt.strftime('%b %Y'), y='number_of_reviews', data=month_reviews, err
plt.title("Trend of the Number of Reviews Per Month", weight = "bold", size = 18)
plt.xticks(size = 10, rotation = 90)
plt.yticks(size = 12)
plt.xlabel("Months", weight = "bold", size = 15)
plt.ylabel("number_of_reviews", weight = "bold", size = 15)
plt.show()
```




**5. Plot a scatter plot to examine the relationship between the price and the number of reviews.**

In [253…

```
plt.figure(figsize = (18, 7))
plt.scatter(x = "price", y = "number_of_reviews", data = df, color = "cyan")
plt.title("The relationship between the Price and the Number of Reviews", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Price", weight = "bold", size = 15)
plt.ylabel("number_of_reviews", weight = "bold", size = 15)
plt.show()
```

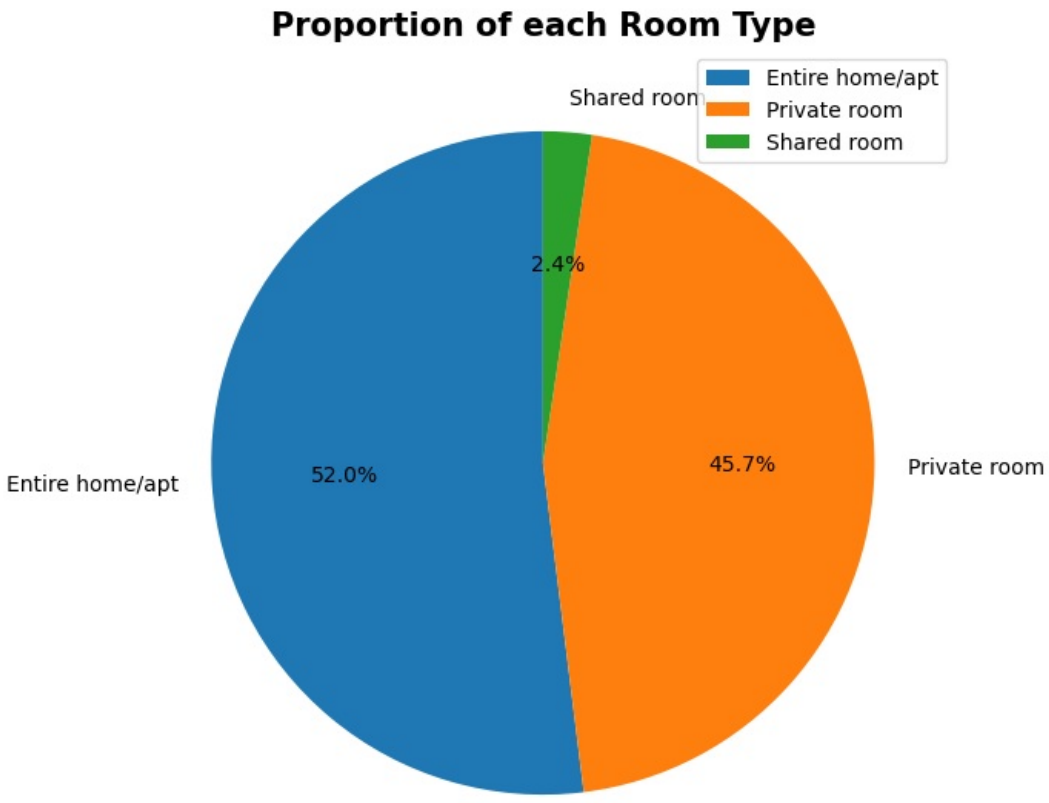

**6. Create a pie chart to display the proportion of each room type.**

In [254…
```
pie_chart = df["room_type"].value_counts().to_frame().reset_index()
pie_chart.columns = ["Room Type", "Counts"]
pie_chart
```

Out[254]:

| | Room Type | Counts |
|---|---|---|
| 0 | Entire home/apt | 25409 |
| 1 | Private room | 22326 |
| 2 | Shared room | 1160 |

In [255…
```
plt.figure(figsize = (18, 7))
plt.pie(pie_chart["Counts"], labels = pie_chart["Room Type"], startangle = 90, autopct ='%1.1f%%')
plt.title('Proportion of each Room Type', weight = "bold", size = 15)
plt.legend()
# Display the chart
plt.show()
```




In [256…
```
df.head(3)
```

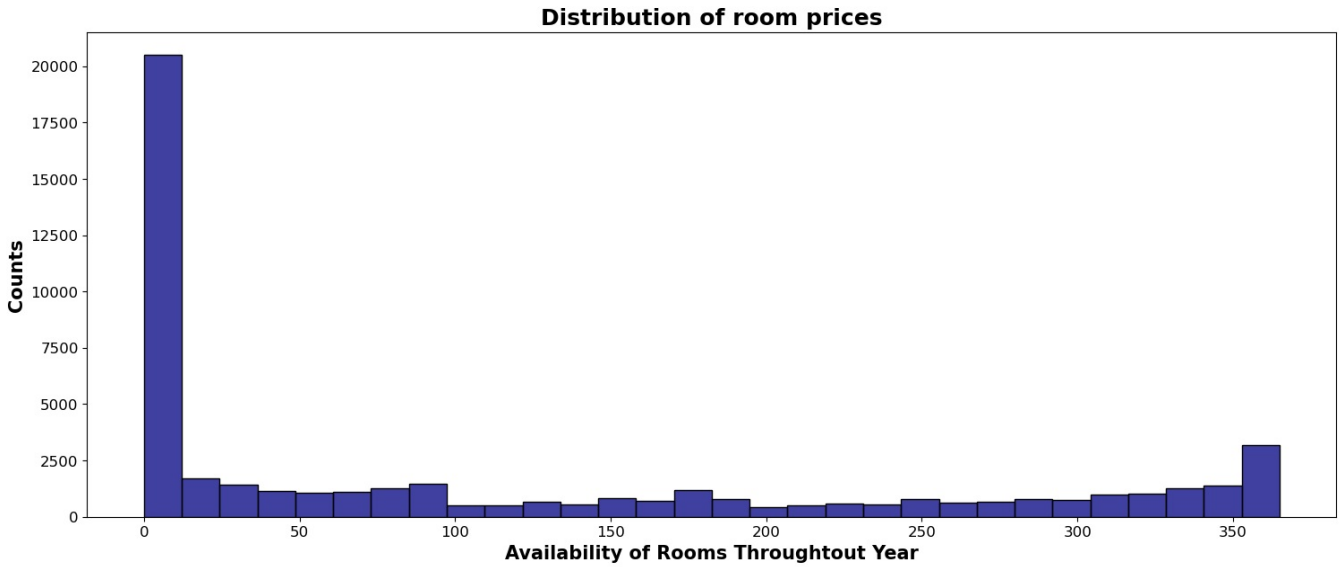
Out[256]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_night |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | |
| 1 | 2595 | Skylit Midtown Castle | 2845 | Jennifer | Manhattan | Midtown | 40.75362 | -73.98377 | Entire home/apt | 225 | |
| 2 | 3647 | THE VILLAGE OF HARLEM....NEW YORK ! | 4632 | Elisabeth | Manhattan | Harlem | 40.80902 | -73.94190 | Private room | 150 | |

**7. Visualize the availability of the rooms using a histogram.**

In [257…

```
plt.figure(figsize = (18, 7))
sns.histplot(x = "availability_365", data = df, color = "navy")
plt.title("Distribution of room prices", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Availability of Rooms Throughtout Year", weight = "bold", size = 15)
plt.ylabel("Counts", weight = "bold", size = 15)
plt.show()
```




**8. Plot a stacked bar chart to compare the availability of rooms for different room types.**

In [258…

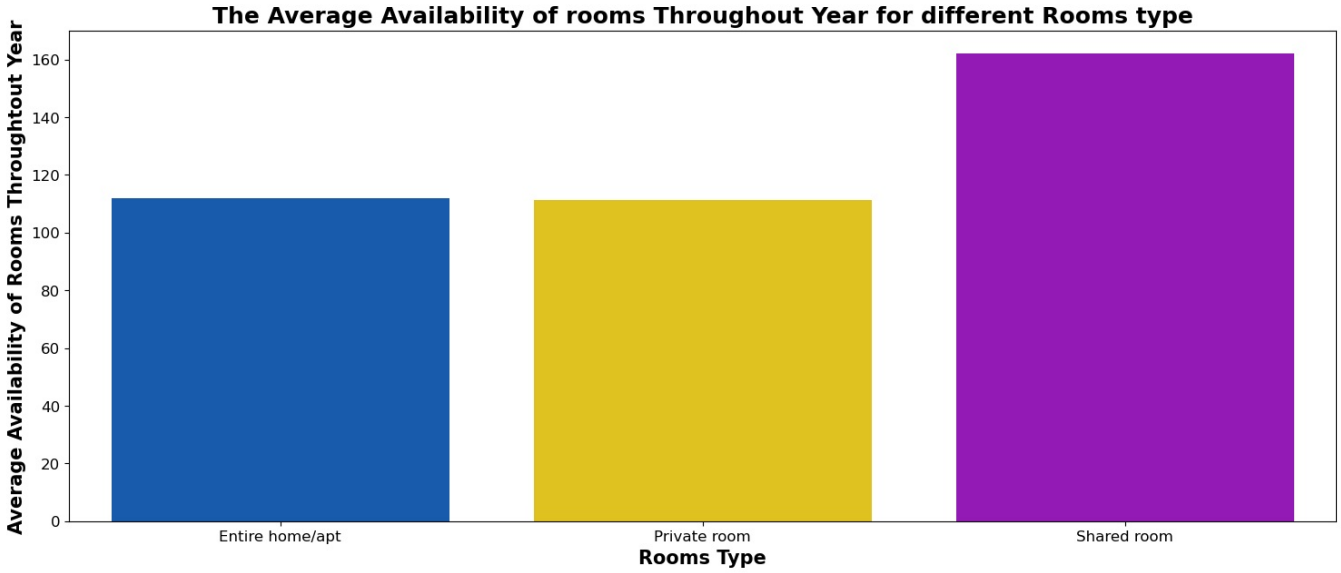
```
rooms_available = df.groupby("room_type")["availability_365"].agg("mean").round(2).to_frame().reset_index()
rooms_available.columns =["Room Type", "Average Available Rooms"]
rooms_available
```

Out[258]:

| | Room Type | Average Available Rooms |
|---|---|---|
| 0 | Entire home/apt | 111.92 |
| 1 | Private room | 111.20 |
| 2 | Shared room | 162.00 |

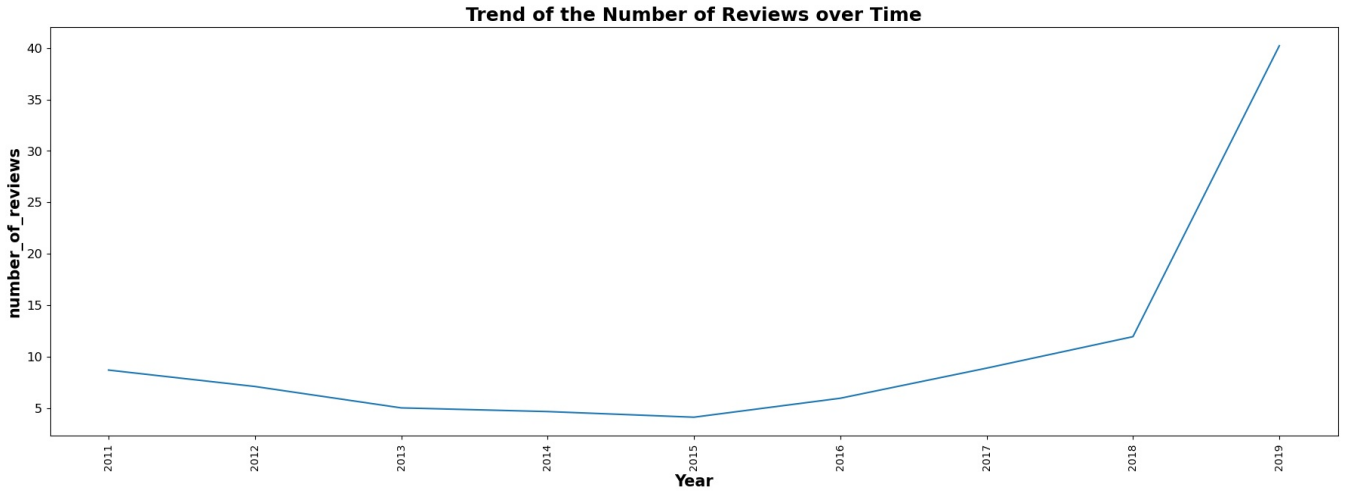
In [259…

```
plt.figure(figsize = (18, 7))
sns.barplot(x ="Room Type", y ="Average Available Rooms", data = rooms_available, palette = "prism")
plt.title("The Average Availability of rooms Throughout Year for different Rooms type", weight = "bold", size =
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Rooms Type", weight = "bold", size = 15)
plt.ylabel("Average Availability of Rooms Throughtout Year", weight = "bold", size = 15)
plt.show()
```




**9. Create a line plot to show the trend of the reviews over time.**

In [275…
```
plt.figure(figsize = (22, 7))
sns.lineplot(x ="Year", y="number_of_reviews", data = month_reviews, errorbar = None)
plt.title("Trend of the Number of Reviews over Time", weight = "bold", size = 18)
plt.xticks(size = 10, rotation = 90)
plt.yticks(size = 12)
plt.xlabel("Year", weight = "bold", size = 15)
plt.ylabel("number_of_reviews", weight = "bold", size = 15)
plt.show()
```



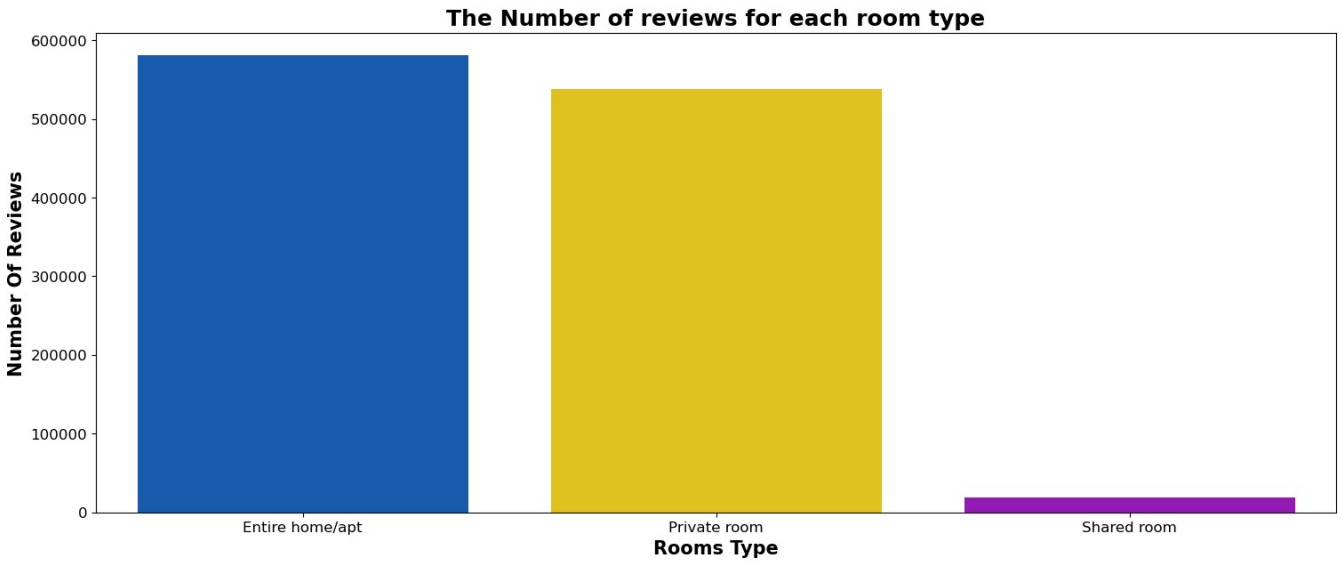

**10. Plot a bar chart to compare the number of reviews for each room type.**

In [260…
```
Bar_chart = df.groupby("room_type")["number_of_reviews"].agg("sum").to_frame().reset_index()
Bar_chart
```

Out[260]:

| | room_type | number_of_reviews |
|---|---|---|
| 0 | Entire home/apt | 580403 |
| 1 | Private room | 538346 |
| 2 | Shared room | 19256 |

In [261…
```
plt.figure(figsize = (18, 7))
sns.barplot(x ="room_type", y ="number_of_reviews", data = Bar_chart, palette = "prism")
plt.title("The Number of reviews for each room type", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Rooms Type", weight = "bold", size = 15)
plt.ylabel("Number Of Reviews", weight = "bold", size = 15)
plt.show()
```



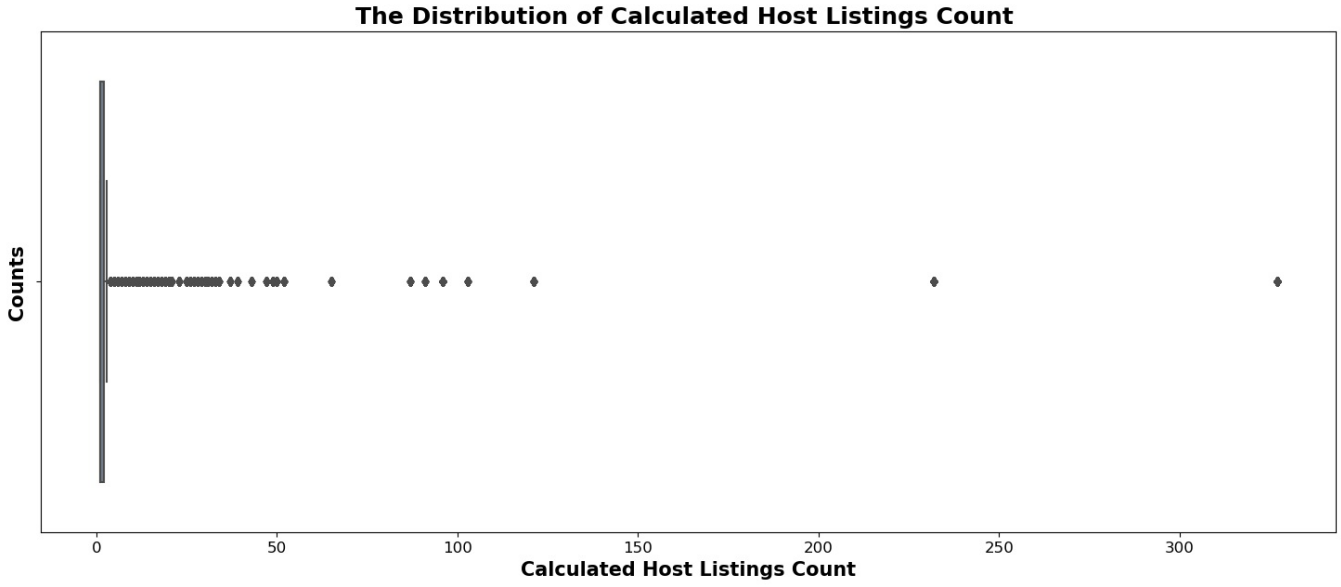

In [262…
```
df.head(1)
```

Out[262]:

| | id | name | host_id | host_name | neighbourhood_group | neighbourhood | latitude | longitude | room_type | price | minimum_nights | number |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2539 | Clean & quiet apt home by the park | 2787 | John | Brooklyn | Kensington | 40.64749 | -73.97237 | Private room | 149 | 1 | |

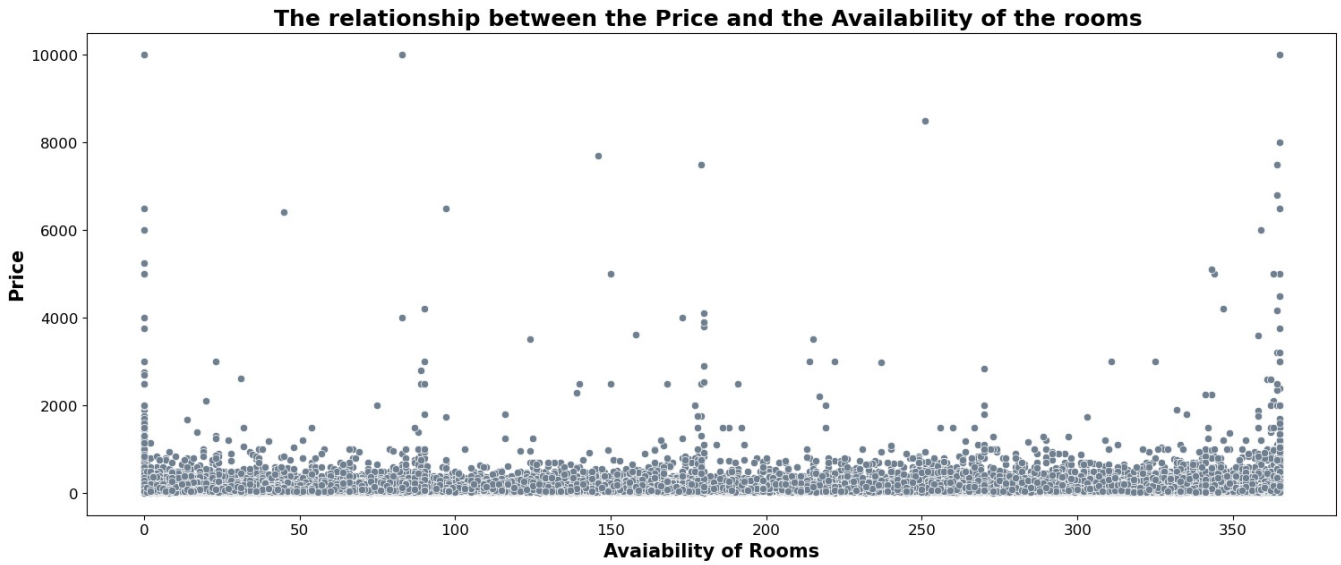
**11. Visualize the distribution of calculated host listings count using a box plot.**

In [263…
```
plt.figure(figsize = (18, 7))
sns.boxplot(x ="calculated_host_listings_count", data =df, color = "slategrey")
plt.title("The Distribution of Calculated Host Listings Count ", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Calculated Host Listings Count", weight = "bold", size = 15)
plt.ylabel("Counts", weight = "bold", size = 15)
plt.show()
```




**12. Create a scatter plot to examine the relationship between the price and the availability of the rooms.**

In [264…
```
plt.figure(figsize = (18, 7))
sns.scatterplot(x ="availability_365", y = "price", data =df, color = "slategrey")
plt.title("The relationship between the Price and the Availability of the rooms ", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Avaiability of Rooms", weight = "bold", size = 15)
plt.ylabel("Price", weight = "bold", size = 15)
plt.show()
```




**13. Plot a line chart to show the trend of the average price over time.**

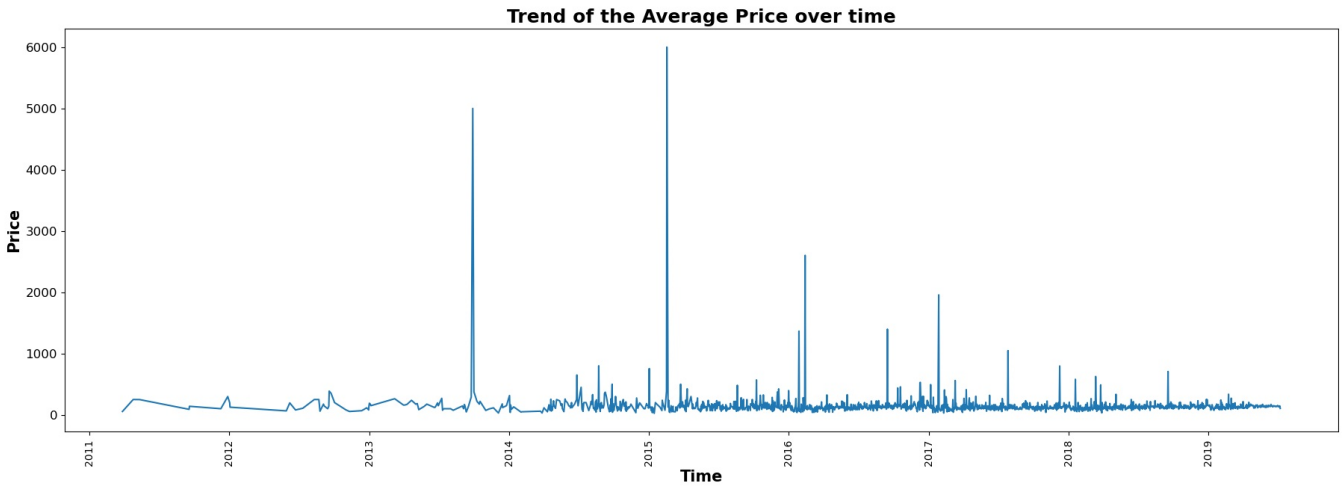
In [279…
```
Line_chart = df.groupby("last_review")["price"].agg("mean").to_frame().reset_index()
Line_chart
```

Out[279]:

| | last_review | price |
|---|---|---|
| **0** | 2011-03-28 | 55.000000 |
| **1** | 2011-04-25 | 250.000000 |
| **2** | 2011-05-12 | 249.000000 |
| **3** | 2011-09-18 | 90.000000 |
| **4** | 2011-09-19 | 140.000000 |
| **...** | ... | ... |
| **1759** | 2019-07-04 | 129.460606 |
| **1760** | 2019-07-05 | 142.301724 |
| **1761** | 2019-07-06 | 139.042478 |
| **1762** | 2019-07-07 | 145.469359 |
| **1763** | 2019-07-08 | 104.887640 |

1764 rows × 2 columns

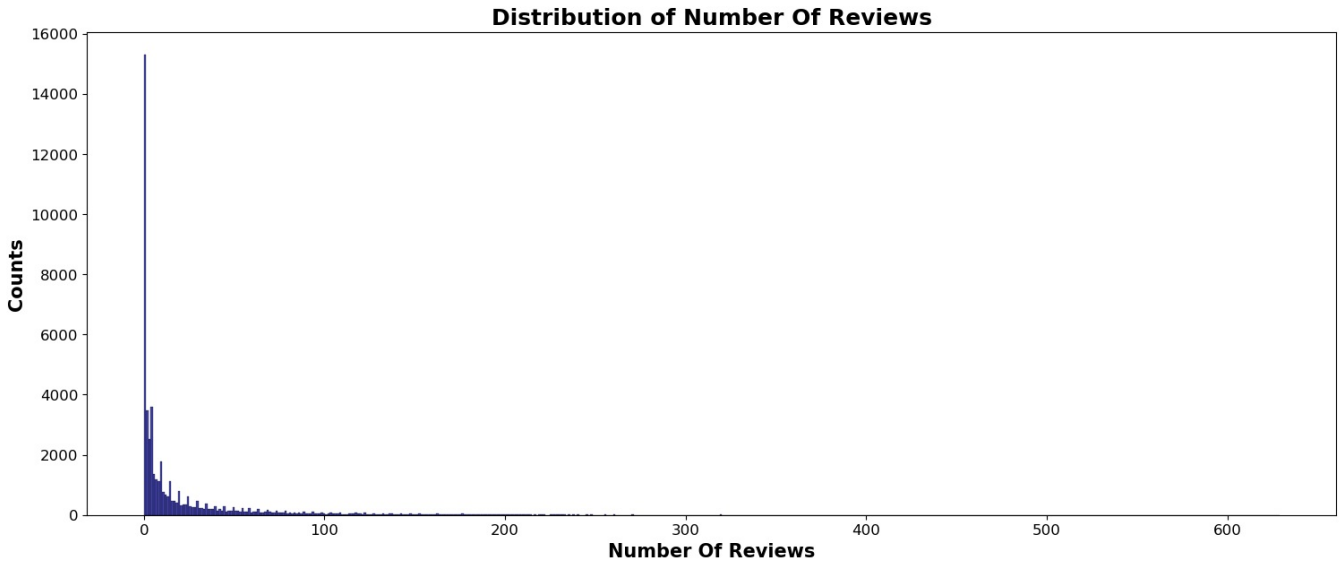
In [280…
```
plt.figure(figsize = (22, 7))
sns.lineplot(x ="last_review", y="price", data = Line_chart, errorbar = None)
plt.title("Trend of the Average Price over time", weight = "bold", size = 18)
plt.xticks(size = 10, rotation = 90)
plt.yticks(size = 12)
plt.xlabel("Time", weight = "bold", size = 15)
plt.ylabel("Price", weight = "bold", size = 15)
plt.show()
```




**14. Visualize the distribution of the number of reviews using a histogram.**

In [265…
```
plt.figure(figsize = (18, 7))
sns.histplot(x ="number_of_reviews", data = df, color = "navy")
plt.title("Distribution of Number Of Reviews", weight = "bold", size = 18)
```

```
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Number Of Reviews", weight = "bold", size = 15)
plt.ylabel("Counts", weight = "bold", size = 15)
plt.show()
```



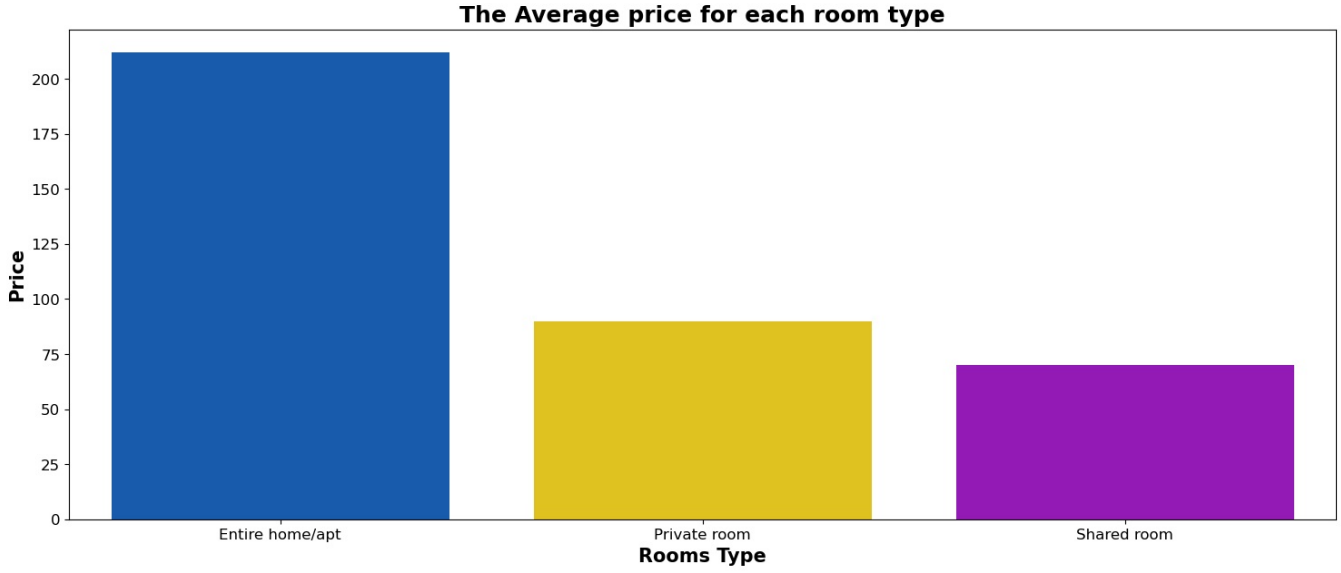

**15. Create a grouped bar chart to compare the average price for each room type.**

In [269…
```
bar_chart = df.groupby("room_type")["price"].agg("mean").to_frame().reset_index().round(2)
bar_chart
```

Out[269]:

| | room_type | price |
|---|---|---|
| 0 | Entire home/apt | 211.79 |
| 1 | Private room | 89.78 |
| 2 | Shared room | 70.13 |

In [272…
```
plt.figure(figsize = (18, 7))
sns.barplot(x ="room_type", y = "price", data = bar_chart, palette = "prism")
plt.title("The Average price for each room type", weight = "bold", size = 18)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.xlabel("Rooms Type", weight = "bold", size = 15)
plt.ylabel("Price", weight = "bold", size = 15)
plt.show()
```




In [ ]:
```
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js