In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import plotly.express as px
```

In [2]:
```
data = pd.read_csv('New_york_dataset.csv')
```

In [4]:
```
data.head()
```

Out[4]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...	last_
0	1.312228e+06	Rental unit in Brooklyn · ★5.0 · 1 bedroom	7130382	Walter	Brooklyn	Clinton Hill	40.683710	-73.964610	Private room	55.0	...	2(
1	4.527754e+07	Rental unit in New York · ★4.67 · 2 bedrooms ·...	51501835	Jeniffer	Manhattan	Hell's Kitchen	40.766610	-73.988100	Entire home/apt	144.0	...	0
2	9.710000e+17	Rental unit in New York · ★4.17 · 1 bedroom · ...	528871354	Joshua	Manhattan	Chelsea	40.750764	-73.994605	Entire home/apt	187.0	...	1
3	3.857863e+06	Rental unit in New York · ★4.64 · 1 bedroom · ...	19902271	John And Catherine	Manhattan	Washington Heights	40.835600	-73.942500	Private room	120.0	...	1
4	4.089661e+07	Condo in New York · ★4.91 · Studio · 1 bed · 1...	61391963	Stay With Vibe	Manhattan	Murray Hill	40.751120	-73.978600	Entire home/apt	85.0	...	0

5 rows × 22 columns

In [5]:
```
data.tail()
```

Out[5]:

	id	name	host_id	host_name	neighbourhood_group	neighbourhood	latitude	longitude	room_type	price	...
20765	2.473690e+07	Rental unit in New York · ★4.75 · 1 bedroom · ...	186680487	Henry D	Manhattan	Lower East Side	40.711380	-73.991560	Private room	45.0	...
20766	2.835711e+06	Rental unit in New York · ★4.46 · 1 bedroom · ...	3237504	Aspen	Manhattan	Greenwich Village	40.730580	-74.000700	Entire home/apt	105.0	...
20767	5.182527e+07	Rental unit in New York · ★4.93 · 1 bedroom · ...	304317395	Jeff	Manhattan	Hell's Kitchen	40.757350	-73.993430	Entire home/apt	299.0	...
20768	7.830000e+17	Rental unit in New York · ★5.0 · 1 bedroom · 1...	163083101	Marissa	Manhattan	Chinatown	40.713750	-73.991470	Entire home/apt	115.0	...
20769	5.660000e+17	Rental unit in Queens · ★4.89 · 1 bedroom · 1 ...	93827372	Glenroy	Queens	Rosedale	40.658874	-73.728651	Private room	102.0	...

5 rows × 22 columns

In [7]:
```
data.shape
```

Out[7]:
```
(20770, 22)
```

In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20770 entries, 0 to 20769
Data columns (total 22 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   id                              20770 non-null  float64
 1   name                            20770 non-null  object
 2   host_id                         20770 non-null  int64
 3   host_name                       20770 non-null  object
 4   neighbourhood_group             20770 non-null  object
 5   neighbourhood                   20763 non-null  object
 6   latitude                        20763 non-null  float64
 7   longitude                       20763 non-null  float64
 8   room_type                       20763 non-null  object
 9   price                           20736 non-null  float64
 10  minimum_nights                  20763 non-null  float64
 11  number_of_reviews               20763 non-null  float64
 12  last_review                     20763 non-null  object
 13  reviews_per_month               20763 non-null  float64
 14  calculated_host_listings_count  20763 non-null  float64
 15  availability_365                20763 non-null  float64
 16  number_of_reviews_ltm           20763 non-null  float64
 17  license                         20770 non-null  object
 18  rating                          20770 non-null  object
 19  bedrooms                        20770 non-null  object
 20  beds                            20770 non-null  int64
 21  baths                           20770 non-null  object
dtypes: float64(10), int64(2), object(10)
memory usage: 3.5+ MB
```

In [12]:
```
data.columns
```

Out[12]:
```
Index(['id', 'name', 'host_id', 'host_name', 'neighbourhood_group',
       'neighbourhood', 'latitude', 'longitude', 'room_type', 'price',
       'minimum_nights', 'number_of_reviews', 'last_review',
       'reviews_per_month', 'calculated_host_listings_count',
       'availability_365', 'number_of_reviews_ltm', 'license', 'rating',
       'bedrooms', 'beds', 'baths'],
      dtype='object')
```

In [13]:
```
data.describe()
```

Out[13]:

	id	host_id	latitude	longitude	price	minimum_nights	number_of_reviews	reviews_per_month	cal
count	2.077000e+04	2.077000e+04	20763.000000	20763.000000	20736.000000	20763.000000	20763.000000	20763.000000	
mean	3.033858e+17	1.749049e+08	40.726821	-73.939179	187.714940	28.558493	42.610605	1.257589	
std	3.901221e+17	1.725657e+08	0.060293	0.061403	1023.245124	33.532697	73.523401	1.904472	
min	2.595000e+03	1.678000e+03	40.500314	-74.249840	10.000000	1.000000	1.000000	0.010000	
25%	2.707260e+07	2.041184e+07	40.684159	-73.980755	80.000000	30.000000	4.000000	0.210000	
50%	4.992852e+07	1.086990e+08	40.722890	-73.949597	125.000000	30.000000	14.000000	0.650000	
75%	7.220000e+17	3.143997e+08	40.763106	-73.917475	199.000000	30.000000	49.000000	1.800000	
max	1.050000e+18	5.504035e+08	40.911147	-73.713650	100000.000000	1250.000000	1865.000000	75.490000	

Let's start data cleaning

In [14]:
```
data.isnull().sum()
```

Out[14]:
```
id                                 0
name                               0
host_id                            0
host_name                          0
neighbourhood_group                0
neighbourhood                      7
latitude                           7
longitude                          7
room_type                          7
price                             34
minimum_nights                     7
number_of_reviews                  7
last_review                        7
reviews_per_month                  7
calculated_host_listings_count     7
availability_365                   7
number_of_reviews_ltm              7
license                            0
rating                             0
bedrooms                           0
beds                               0
baths                              0
dtype: int64
```

In [16]:
```
data.dropna(inplace=True)
data.isnull().sum()
```

Out[16]:
```
id                                 0
name                               0
host_id                            0
host_name                          0
neighbourhood_group                0
neighbourhood                      0
latitude                           0
longitude                          0
room_type                          0
price                              0
minimum_nights                     0
number_of_reviews                  0
last_review                        0
reviews_per_month                  0
calculated_host_listings_count     0
availability_365                   0
number_of_reviews_ltm              0
license                            0
rating                             0
bedrooms                           0
beds                               0
baths                              0
dtype: int64
```

In [19]:
```
data.duplicated().sum()
```

Out[19]:
```
np.int64(12)
```

In [20]:
```
data.drop_duplicates(inplace=True)
data.duplicated().sum()
```

Out[20]:
```
np.int64(0)
```

In [21]:
```
data.dtypes
```

Out[21]:
```
id                                 float64
name                                object
host_id                              int64
host_name                           object
neighbourhood_group                 object
neighbourhood                       object
latitude                           float64
longitude                          float64
room_type                           object
price                              float64
minimum_nights                     float64
number_of_reviews                  float64
last_review                         object
reviews_per_month                  float64
calculated_host_listings_count     float64
availability_365                   float64
number_of_reviews_ltm              float64
license                             object
rating                              object
bedrooms                            object
beds                                 int64
baths                               object
dtype: object
```

In [22]:
```
data['id'] = data['id'].astype(object)
data.dtypes
```

Out[22]:
```
id                                 object
name                                object
host_id                              int64
host_name                           object
neighbourhood_group                 object
neighbourhood                       object
latitude                           float64
longitude                          float64
room_type                           object
price                              float64
minimum_nights                     float64
number_of_reviews                  float64
last_review                         object
reviews_per_month                  float64
calculated_host_listings_count     float64
availability_365                   float64
number_of_reviews_ltm              float64
license                             object
rating                              object
bedrooms                            object
beds                                 int64
baths                               object
dtype: object
```

In [23]:
```
data['host_id'] = data['host_id'].astype(object)
data.dtypes
```

Out[23]:
```
id                                 object
name                                object
host_id                             object
host_name                           object
neighbourhood_group                 object
neighbourhood                       object
latitude                           float64
longitude                          float64
room_type                           object
price                              float64
minimum_nights                     float64
number_of_reviews                  float64
last_review                         object
reviews_per_month                  float64
calculated_host_listings_count     float64
availability_365                   float64
number_of_reviews_ltm              float64
license                             object
rating                              object
bedrooms                            object
beds                                 int64
baths                               object
dtype: object
```

In [25]:
```
data.columns = data.columns.str.strip().str.lower().str.replace(' ', '_')
data['last_review'] = pd.to_datetime(data['last_review'], errors = 'coerce')
data['rating'] = pd.to_numeric(data['rating'], errors = 'coerce')
data['bedrooms'] = pd.to_numeric(data['bedrooms'], errors='coerce')
data['baths'] = pd.to_numeric(data['baths'], errors='coerce')
data['price'] = pd.to_numeric(data['price'], errors='coerce')
```

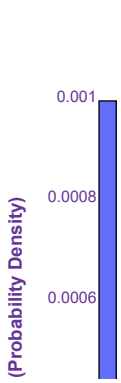
In [26]:
```
df = data.dropna(subset=['latitude', 'longitude', 'price'])
```

Room Type Distribution

In [28]:
```
fig1 = px.pie(
    df,
    names='room_type',
    title='Distribution of Room Types'
)
fig1.show()
```

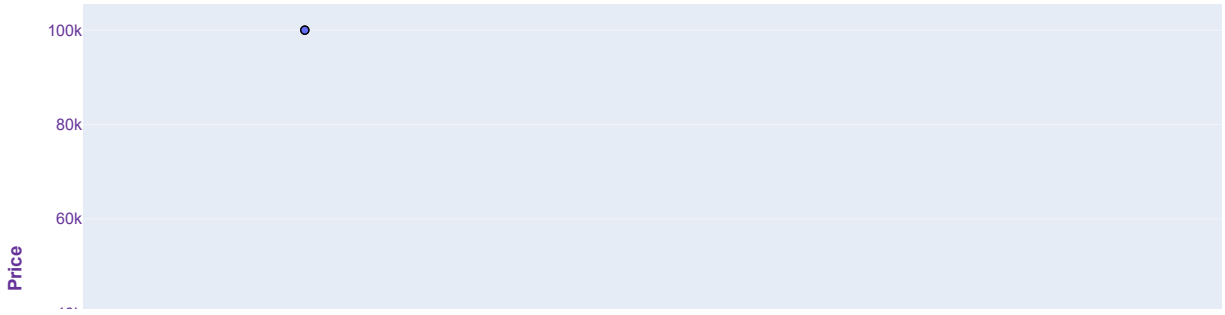
Price Distribution

In [70]:
```
fig2 = px.histogram(
    df,
    x = 'price',
    nbins= 100,
    histnorm='probability density',
    title = 'Price Distribution',
    hover_data= {'price' : ':.2f'},
    labels= {'price' : 'price'}
)
fig2.update_traces(
    marker = dict(line = dict(width =1 , color = 'black')),
    selector = dict(type = 'histogram')
)
fig2.update_layout(
    xaxis_title='<b>Price </b>',
    yaxis_title='<b>Frequency (Probability Density)</b>',
    bargap=0.1,
    plot_bgcolor='white',
    paper_bgcolor='white',
    font=dict(family="Arial, sans-serif", size=12, color="RebeccaPurple"),
    title_x=0.5,
    margin=dict(l=40, r=40, t=80, b=40),
    hovermode="x unified"
)
fig2.show()
```



In [ ]:
```
# Price per Room Type
```

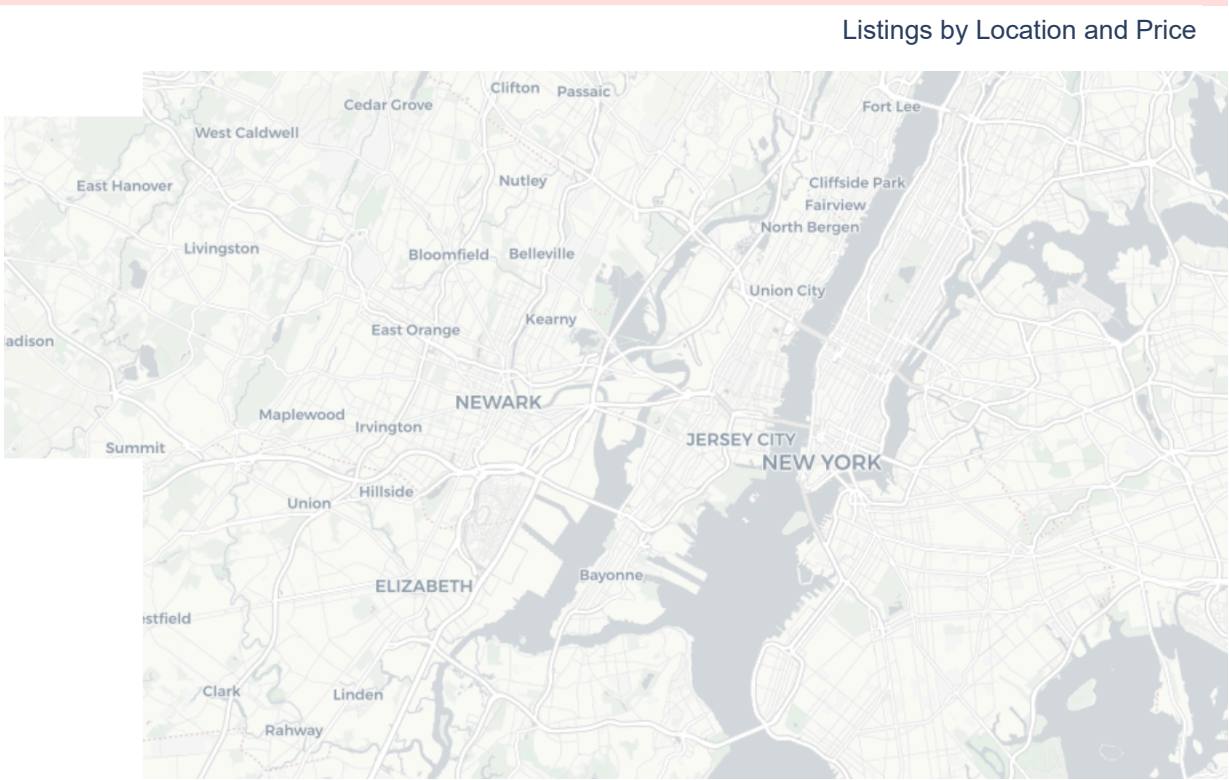
In [41]:
```
fig3 = px.box(
    df,
    x='room_type',
    y='price',
    color='room_type',
    title='Price per Room Type'
)
fig3.update_traces(
    marker = dict(line =dict(color = 'black', width = 1)),
    selector = dict(type ='box')
)
fig3.update_layout(
    xaxis_title='<b>Room Type</b>',
    yaxis_title='<b>Price </b>',
    bargap = 0.2,
    paper_bgcolor = 'white',
    # plot_bgcolor = 'lightblue',
    font=dict(family="Arial, sans-serif", size=12, color="RebeccaPurple"),
    title_x=0.5,
    margin=dict(l=40, r=40, t=80, b=40),
    hovermode = 'x unified'
)
fig3.show()
```



In [46]:
```
fig4 = px.scatter_mapbox(
    df,
    lat='latitude',
    lon='longitude',
    color='price',
    size='price',
    hover_name='neighbourhood',
    color_continuous_scale='Viridis',
    size_max=15,
    zoom=10,
    height=600,
    title='Listings by Location and Price'
)
fig4.update_layout(
    mapbox_style='carto-positron',
    margin=dict(l=20, r=20, t=50, b=20),
    title_font=dict(size=20, family='Arial'),
    title_x=0.5
)
fig4.show()
```

```
C:\Users\prita\AppData\Local\Temp\ipykernel_11088\1014570141.py:1: DeprecationWarning:

*scatter_mapbox* is deprecated! Use *scatter_map* instead. Learn more at: https://plotly.com/python/mapbox-to-maplibre/
```

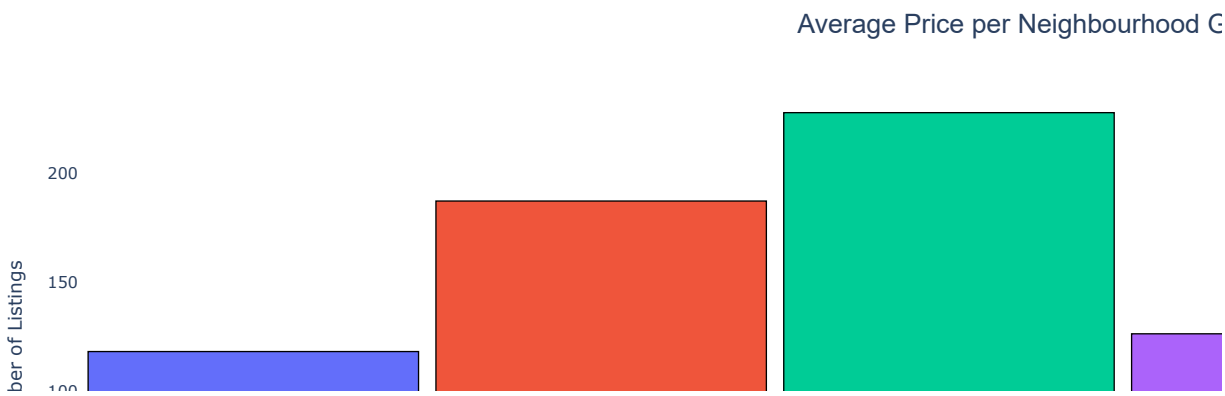


Average Price per Neighbourhood Group

In [51]:
```
avg_price_neigh = df.groupby('neighbourhood_group')['price'].mean().reset_index()

fig5 = px.bar(
    avg_price_neigh,
    x='neighbourhood_group',
    y='price',
    color='neighbourhood_group',
    title='Average Price per Neighbourhood Group'
)
fig5.update_traces(
    marker = dict(line=dict(width =1 , color = 'black')),
    selector = dict(type='bar')
)
fig5.update_layout(
     title_font=dict(size=20, family='Arial'),
    yaxis_title='Number of Listings',
    xaxis_title='Price',
    bargap=0.05,
    plot_bgcolor='white',
    paper_bgcolor='white',
    title_x=0.5,
    margin=dict(l=40, r=40, t=80, b=40),
```

```
    hovermode="x unified"
)
fig5.show()
```



Number of Reviews per Room Type

In [65]:
```
fig6 = px.bar(
    df.groupby('room_type')['number_of_reviews'].sum().reset_index(),
    x='room_type', y='number_of_reviews',
    title='Total Number of Reviews per Room Type',
    color='room_type'
)
fig6.show()
```