```
!pip install pymongo
```

```
Collecting pymongo
  Downloading pymongo-4.6.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (677 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 677.2/677.2 kB 9.8 MB/s eta 0:00:00
Collecting dnspython<3.0.0,>=1.16.0 (from pymongo)
  Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.7/307.7 kB 13.5 MB/s eta 0:00:00
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.6.1 pymongo-4.6.2
```

```
import pandas as pd
import numpy as np
import pymongo
import seaborn as sns
import plotly.express as px
import matplotlib.pyplot as plt
from wordcloud import STOPWORDS, WordCloud
```

```
client = pymongo.MongoClient('mongodb+srv://abhilashgnagar22:MxXIvbLjVJ5vNZLX@cluster0.h2r55k6.mongodb.net/?retryWrites=true&w=majority&a
db = client.sample_airbnb
col = db.listingsAndReviews
```

```
rel_data = []
for i in col.find():
    data = dict(Id = i['_id'],
                Listing_url = i['listing_url'],
                Name = i.get('name'),
                Description = i['description'],
                House_rules = i.get('house_rules'),
                Property_type = i['property_type'],
                Room_type = i['room_type'],
                Bed_type = i['bed_type'],
                Min_nights = int(i['minimum_nights']),
                Max_nights = int(i['maximum_nights']),
                Cancellation_policy = i['cancellation_policy'],
                Accomodates = i['accommodates'],
                Total_bedrooms = i.get('bedrooms'),
                Total_beds = i.get('beds'),
                Availability_365 = i['availability']['availability_365'],
                Price = i['price'],
                Security_deposit = i.get('security_deposit'),
                Cleaning_fee = i.get('cleaning_fee'),
                Extra_people = i['extra_people'],
                Guests_included= i['guests_included'],
                No_of_reviews = i['number_of_reviews'],
                Review_scores = i['review_scores'].get('review_scores_rating'),
                Amenities = ', '.join(i['amenities']),
                Host_id = i['host']['host_id'],
                Host_name = i['host']['host_name'],
                Street = i['address']['street'],
                Country = i['address']['country'],
                Country_code = i['address']['country_code'],
                Location_type = i['address']['location']['type'],
                Longitude = i['address']['location']['coordinates'][0],
                Latitude = i['address']['location']['coordinates'][1],
                Is_location_exact = i['address']['location']['is_location_exact']
    )
    rel_data.append(data)
```

```
df = pd.DataFrame(rel_data)
df.head()
```

	Id	Listing_url	Name	Description	House_rules	Property_type	Room_type	Bed_type	Min_nights
0	10006546	https://www.airbnb.com/rooms/10006546	Ribeira Charming Duplex	Fantastic duplex apartment with three bedrooms...	Make the house your home...	House	Entire home/apt	Real Bed	2
1	10009999	https://www.airbnb.com/rooms/10009999	Horto flat with small garden	One bedroom + sofa-bed in quiet and bucolic ne...	I just hope the guests treat the space as they...	Apartment	Entire home/apt	Real Bed	2
2	1001265	https://www.airbnb.com/rooms/1001265	Ocean View Waikiki Marina w/prkg	A short distance from Honolulu's billion dolla...	The general welfare and well being of all the ...	Condominium	Entire home/apt	Real Bed	3
3	10021707	https://www.airbnb.com/rooms/10021707	Private Room in Bushwick	Here exists a very cozy room for rent in a sha...		Apartment	Private room	Real Bed	14
4	10030955	https://www.airbnb.com/rooms/10030955	Apt Linda Vista Lagoa - Rio	Quarto com vista para a Lagoa Rodrigo de Freit...		Apartment	Private room	Real Bed	1

5 rows × 32 columns

```
# checking Data types
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5555 entries, 0 to 5554
Data columns (total 32 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Id                   5555 non-null   object 
 1   Listing_url          5555 non-null   object 
 2   Name                 5555 non-null   object 
 3   Description          5555 non-null   object 
 4   House_rules          5555 non-null   object 
 5   Property_type        5555 non-null   object 
 6   Room_type            5555 non-null   object 
 7   Bed_type             5555 non-null   object 
 8   Min_nights           5555 non-null   int64  
 9   Max_nights           5555 non-null   int64  
 10  Cancellation_policy  5555 non-null   object 
 11  Accomodates          5555 non-null   int64  
 12  Total_bedrooms       5550 non-null   float64
 13  Total_beds           5542 non-null   float64
 14  Availability_365     5555 non-null   int64  
 15  Price                5555 non-null   object 
 16  Security_deposit     3471 non-null   object 
 17  Cleaning_fee         4024 non-null   object 
 18  Extra_people         5555 non-null   object 
 19  Guests_included      5555 non-null   object 
 20  No_of_reviews        5555 non-null   int64  
 21  Review_scores        4081 non-null   float64
 22  Amenities            5555 non-null   object 
 23  Host_id              5555 non-null   object 
 24  Host_name            5555 non-null   object 
 25  Street               5555 non-null   object 
 26  Country              5555 non-null   object 
 27  Country_code         5555 non-null   object 
 28  Location_type        5555 non-null   object 
 29  Longitude            5555 non-null   float64
 30  Latitude             5555 non-null   float64
 31  Is_location_exact    5555 non-null   bool   
dtypes: bool(1), float64(5), int64(5), object(21)
memory usage: 1.3+ MB
```

```
# The below features are in Decimal128 type hence changing it to relevant data types
df.Price = df.Price.astype(str).astype(float)
df.Security_deposit = df.Security_deposit[~df.Security_deposit.isna()].astype(str).astype(float)
df.Cleaning_fee = df.Cleaning_fee[~df.Cleaning_fee.isna()].astype(str).astype(float)
df.Extra_people = df.Extra_people.astype(str).astype(float)
df.Guests_included = df.Guests_included.astype(str).astype(float)
df.Review_scores = df.Review_scores.astype('Int64')
df.isna().sum()
```

```
     Id                        0
     Listing_url               0
     Name                      0
     Description               0
     House_rules               0
     Property_type             0
     Room_type                 0
     Bed_type                  0
     Min_nights                0
     Max_nights                0
     Cancellation_policy       0
     Accomodates               0
     Total_bedrooms            5
     Total_beds               13
     Availability_365          0
     Price                     0
     Security_deposit       2084
     Cleaning_fee           1531
     Extra_people              0
     Guests_included           0
     No_of_reviews             0
     Review_scores          1474
     Amenities                 0
     Host_id                   0
     Host_name                 0
     Street                    0
     Country                   0
     Country_code              0
     Location_type             0
     Longitude                 0
     Latitude                  0
     Is_location_exact         0
     dtype: int64
```

```
# Filling Total bedrooms with mode
df.Total_bedrooms.fillna(df.Total_bedrooms.mode()[0],inplace=True)
# Filling Total beds with median because data has outliers
df.Total_beds.fillna(df.Total_beds.median(),inplace=True)
df.Security_deposit.fillna(df.Security_deposit.median(),inplace=True)
df.Cleaning_fee.fillna(df.Cleaning_fee.median(),inplace=True)
df.Review_scores.fillna(df.Review_scores.median(),inplace=True)
```

```
# Filling Empty values in Description and House rules columns
df.Description.replace(to_replace='',value='No Description Provided',inplace=True)
df.House_rules.replace(to_replace='',value='No House rules Provided',inplace=True)
df.Amenities.replace(to_replace='',value='Not Available',inplace=True)
```

```
df.isna().sum()
```

```
     Id                     0
     Listing_url            0
     Name                   0
     Description            0
     House_rules            0
     Property_type          0
     Room_type              0
     Bed_type               0
     Min_nights             0
     Max_nights             0
     Cancellation_policy    0
     Accomodates            0
     Total_bedrooms         0
     Total_beds             0
     Availability_365       0
     Price                  0
     Security_deposit       0
     Cleaning_fee           0
     Extra_people           0
     Guests_included        0
     No_of_reviews          0
     Review_scores          0
     Amenities              0
     Host_id                0
     Host_name              0
     Street                 0
     Country                0
     Country_code           0
     Location_type          0
     Longitude              0
     Latitude               0
     Is_location_exact      0
     dtype: int64
```

```
# Checking Duplicate records
df[df.duplicated()]
```

Id	Listing_url	Name	Description	House_rules	Property_type	Room_type	Bed_type	Min_nights	Max_nights	...	Amenities	Host_i

0 rows × 32 columns

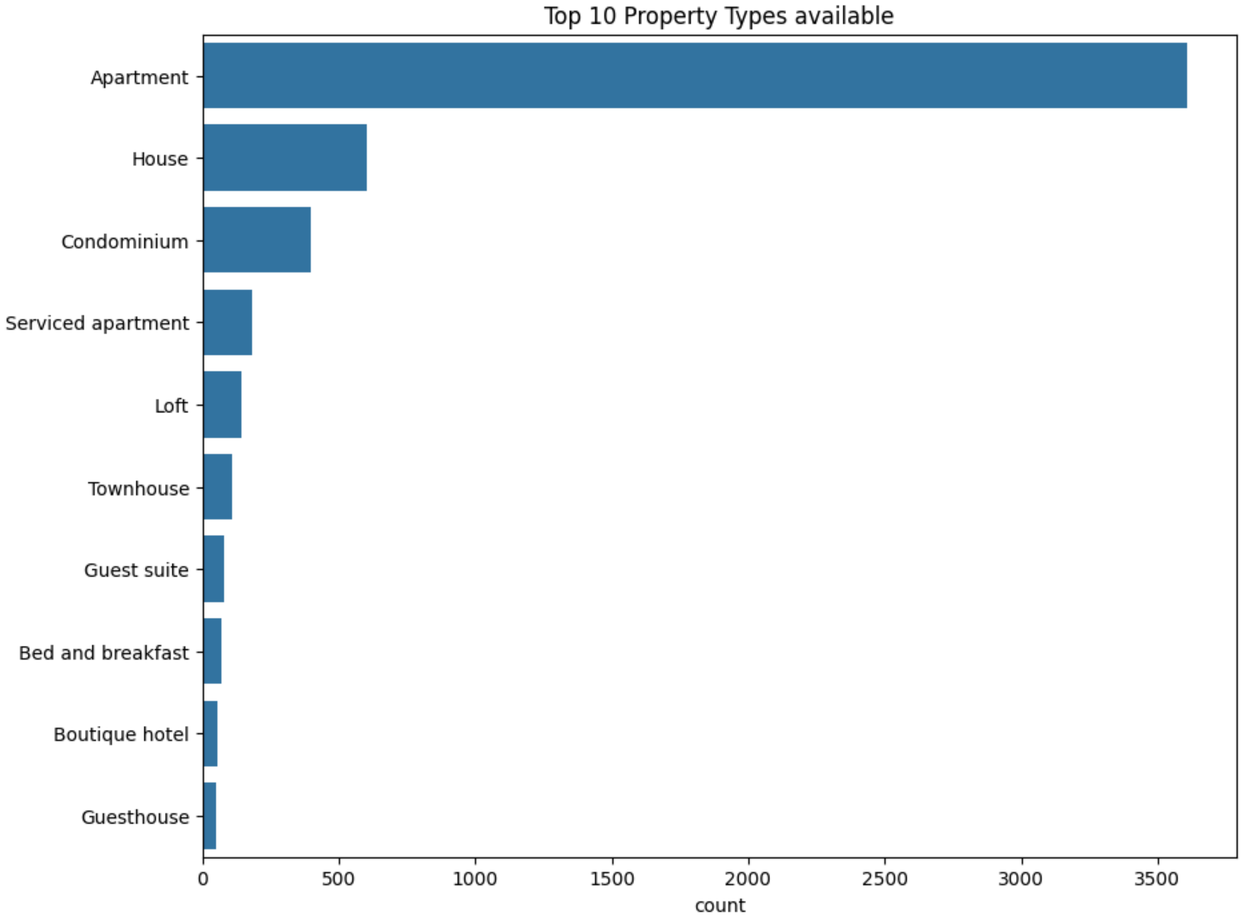
```
# Name Column has empty values and some duplicates hence dropping them
df.drop(labels=list(df[df.Name.duplicated(keep=False)].index),inplace=True)
```

```
df.reset_index(drop=True,inplace=True)
```

```
# Converting dataframe to csv file and saving it
df.to_csv('Airbnb_data.csv',index=False)
```

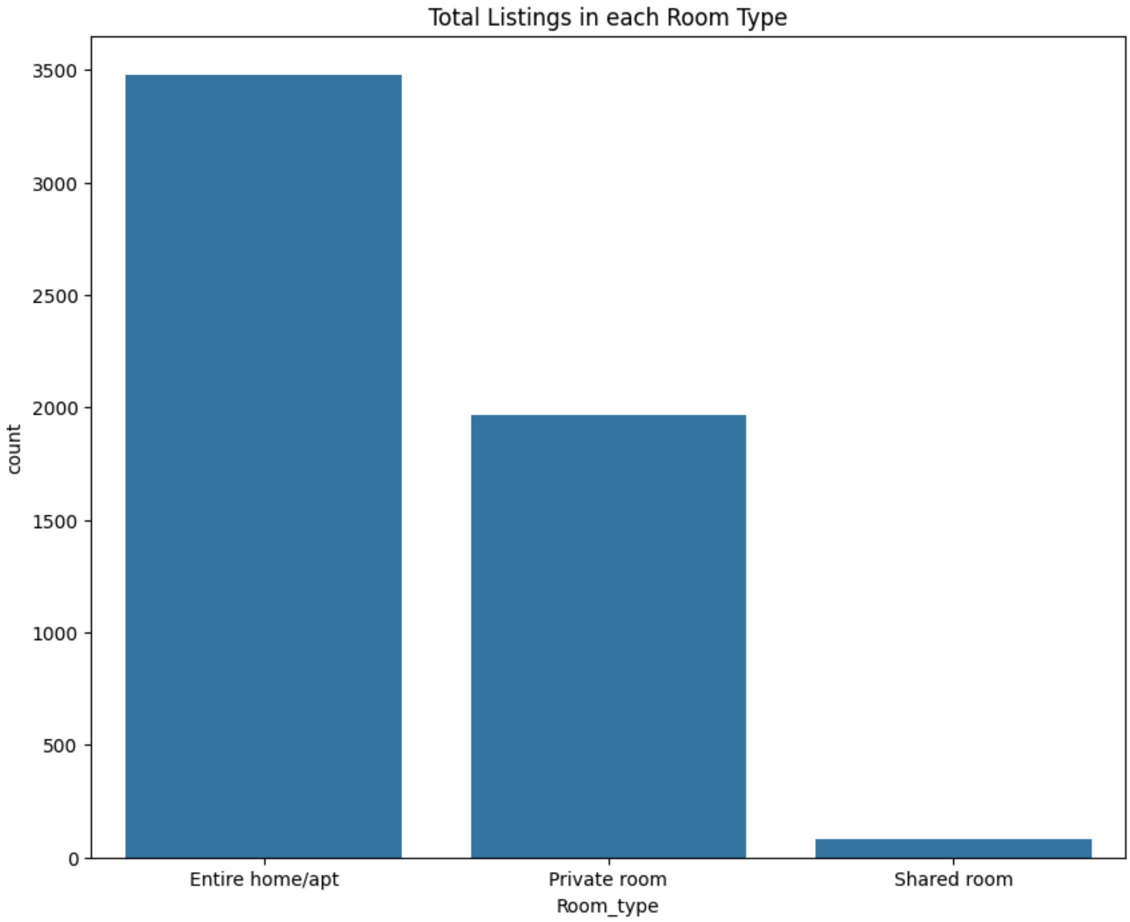
```
plt.figure(figsize=(10,8))
ax = sns.countplot(data=df,y=df.Property_type.values,order=df.Property_type.value_counts().index[:10])
ax.set_title("Top 10 Property Types available")
```

```
Text(0.5, 1.0, 'Top 10 Property Types available')
```



```
plt.figure(figsize=(10,8))
ax = sns.countplot(data=df,x=df.Room_type)
ax.set_title("Total Listings in each Room Type")
```

```
Text(0.5, 1.0, 'Total Listings in each Room Type')
```

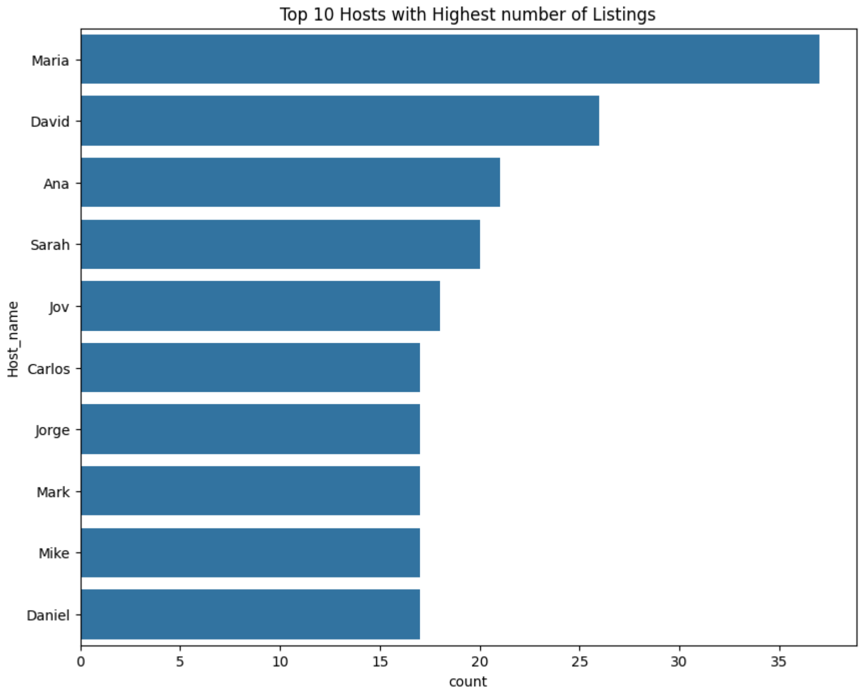


```
# top 10 Hosts with Highest number of listings
df.Host_name.value_counts()
```

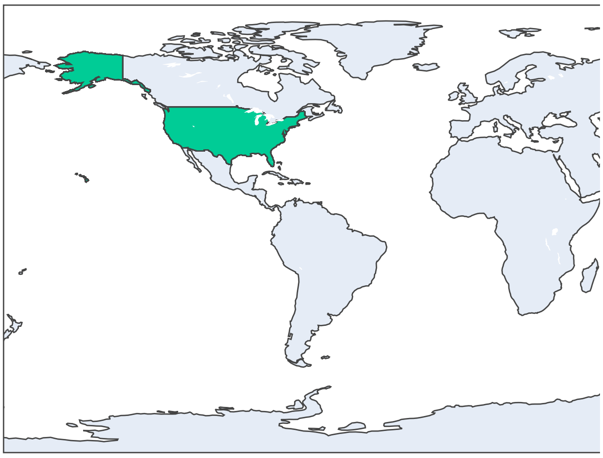
```
Maria             37
David             26
Ana               21
Sarah             20
Jov               18
                  ..
Five Seven Nine    1
Yeimy              1
Isa                1
Allure Villas      1
Ana&Gonçalo        1
Name: Host_name, Length: 3134, dtype: int64
```

```
plt.figure(figsize=(10,8))
ax = sns.countplot(data=df,y=df.Host_name,order=df.Host_name.value_counts().index[:10])
ax.set_title("Top 10 Hosts with Highest number of Listings")
```

```
Text(0.5, 1.0, 'Top 10 Hosts with Highest number of Listings')
```

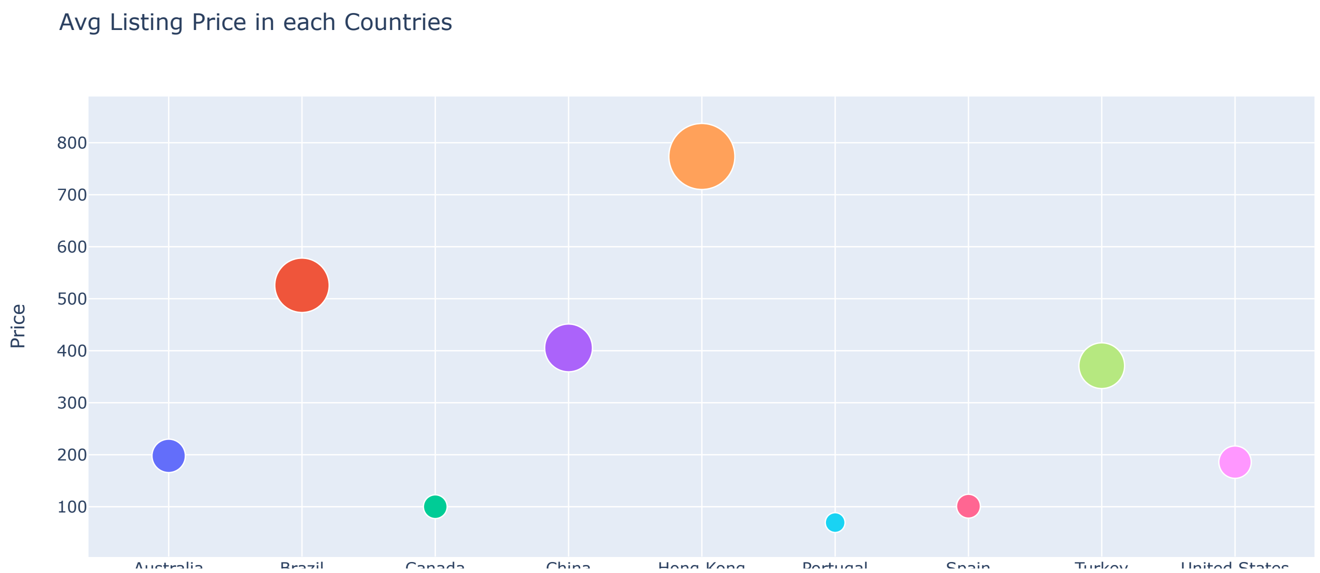


```
fig = px.choropleth(data_frame=df,
                    locations='Country_code',
                    color='Country',
                    locationmode='country names')
fig.show()
```

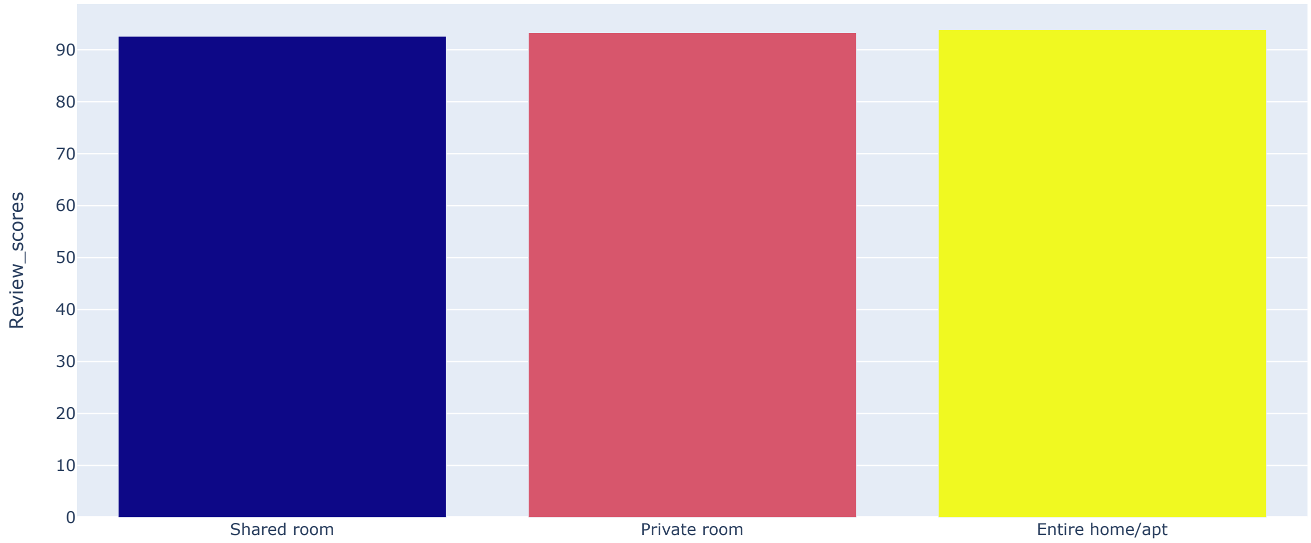


```
country_df = df.groupby('Country',as_index=False)['Price'].mean()
```

```
fig = px.scatter(data_frame=country_df,
           x='Country',y='Price',
           color='Country',
           size='Price',
           opacity=1,
           size_max=35,
           title='Avg Listing Price in each Countries')
fig.show()
```



```
rev_df = df.groupby('Room_type',as_index=False)['Review_scores'].mean().sort_values(by='Review_scores')
fig = px.bar(data_frame=rev_df,x='Room_type',y='Review_scores',color='Review_scores')
fig.show()
```


Review_scores
Shared room
Private room
Entire home/apt