Only for Air_Bnb case study analysis and sorting and reviewing the data.

"For the AirBnB case study analysis, we will analyze the dataset to find the lowest and cheapest travel options in European countries."

In [6]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

In [13]:
```
import zipfile
import os
import shutil

# Path to the ZIP file
zip_path = "D:\\aman_new\\Listings.csv"
extract_folder = "D:\\aman_new\\Listings.csv"
```

In [16]:
```
df = pd.read_csv("D:\\aman_new\\Listings.csv\\Listings.csv",encoding='latin1')
df.head()
```

Out[16]:

	listing_id	name	host_id	host_since	host_location	host_response_time	host_res
0	281420	Beautiful Flat in le Village Montmartre, Paris	1466919	2011-12-03	Paris, Ile-de-France, France	NaN	
1	3705183	39 mÃ□Â² Paris (Sacre CÃ□â□□ur)	10328771	2013-11-29	Paris, Ile-de-France, France	NaN	
2	4082273	Lovely apartment with Terrace, 60m2	19252768	2014-07-31	Paris, Ile-de-France, France	NaN	
3	4797344	Cosy studio (close to Eiffel tower)	10668311	2013-12-17	Paris, Ile-de-France, France	NaN	
4	4823489	Close to Eiffel Tower - Beautiful flat : 2 rooms	24837558	2014-12-14	Paris, Ile-de-France, France	NaN	

5 rows × 33 columns

In [17]:
```
df.shape
```

Out[17]:
```
(279712, 33)
```

In [19]:
```
df.describe()
```

Out[19]:

	listing_id	host_id	host_response_rate	host_acceptance_rate	host_total_list
count	2.797120e+05	2.797120e+05	150930.000000	166625.000000	279
mean	2.638196e+07	1.081658e+08	0.865939	0.827168	
std	1.442576e+07	1.108570e+08	0.283744	0.289202	
min	2.577000e+03	1.822000e+03	0.000000	0.000000	
25%	1.384462e+07	1.720656e+07	0.900000	0.780000	
50%	2.767098e+07	5.826911e+07	1.000000	0.980000	
75%	3.978485e+07	1.832853e+08	1.000000	1.000000	
max	4.834353e+07	3.901874e+08	1.000000	1.000000	7

In [20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 279712 entries, 0 to 279711
Data columns (total 33 columns):
 #   Column                        Non-Null Count   Dtype  
---  ------                        --------------   -----  
 0   listing_id                    279712 non-null  int64  
 1   name                          279537 non-null  object 
 2   host_id                       279712 non-null  int64  
 3   host_since                    279547 non-null  object 
 4   host_location                 278872 non-null  object 
 5   host_response_time            150930 non-null  object 
 6   host_response_rate            150930 non-null  float64
 7   host_acceptance_rate          166625 non-null  float64
 8   host_is_superhost             279547 non-null  object 
 9   host_total_listings_count     279547 non-null  float64
 10  host_has_profile_pic          279547 non-null  object 
 11  host_identity_verified        279547 non-null  object 
 12  neighbourhood                 279712 non-null  object 
 13  district                      37012 non-null   object 
 14  city                          279712 non-null  object 
 15  latitude                      279712 non-null  float64
 16  longitude                     279712 non-null  float64
 17  property_type                 279712 non-null  object 
 18  room_type                     279712 non-null  object 
 19  accommodates                  279712 non-null  int64  
 20  bedrooms                      250277 non-null  float64
 21  amenities                     279712 non-null  object 
 22  price                         279712 non-null  int64  
 23  minimum_nights                279712 non-null  int64  
 24  maximum_nights                279712 non-null  int64  
 25  review_scores_rating          188307 non-null  float64
 26  review_scores_accuracy        187999 non-null  float64
 27  review_scores_cleanliness     188047 non-null  float64
 28  review_scores_checkin         187941 non-null  float64
 29  review_scores_communication   188025 non-null  float64
 30  review_scores_location        187937 non-null  float64
 31  review_scores_value           187927 non-null  float64
 32  instant_bookable              279712 non-null  object 
dtypes: float64(13), int64(6), object(14)
memory usage: 70.4+ MB
```

In [21]:
```
# we have check the name of the columns
df.columns
```

Out[21]:
```
Index(['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', 'bedrooms', 'amenities', 'price',
       'minimum_nights', 'maximum_nights', 'review_scores_rating',
       'review_scores_accuracy', 'review_scores_cleanliness',
       'review_scores_checkin', 'review_scores_communication',
       'review_scores_location', 'review_scores_value', 'instant_bookable'],
      dtype='object')
```

In [22]:
```
df.groupby('city')['instant_bookable'].count()
```

Out[22]:
```
city
Bangkok          19361
Cape Town        19086
Hong Kong         7087
Istanbul         24519
Mexico City      20065
New York         37012
Paris            64690
Rio de Janeiro   26615
Rome             27647
Sydney           33630
Name: instant_bookable, dtype: int64
```

1. can you spot any major differences in the Airbnb market between cities?

In [24]:
```
sorted_df = df.groupby('city')['instant_bookable'].count().sort_values(ascending =
```

In [25]:
```
print(sorted_df)
```

```
city
Paris            64690
New York         37012
Sydney           33630
Rome             27647
Rio de Janeiro   26615
Istanbul         24519
Mexico City      20065
Bangkok          19361
Cape Town        19086
Hong Kong         7087
Name: instant_bookable, dtype: int64
```

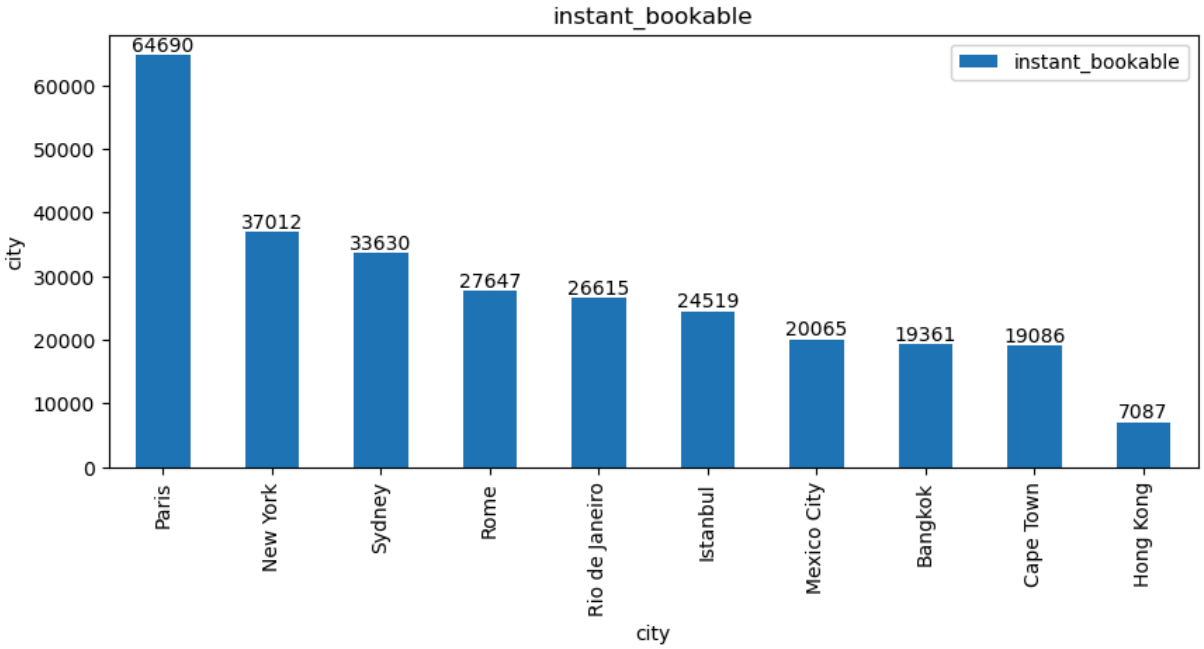
In [31]:
```
sorted_df.head(30).to_frame().plot(kind='bar',figsize=(10,4))

for i, v in enumerate(sorted_df):
    plt.text(i, v, str(v), ha='center', va='bottom')

plt.ylabel('city')
```

```
plt.title('instant_bookable')
```

Out[31]: Text(0.5, 1.0, 'instant_bookable')



2.Which attributes have the biggest influence on price?

In [33]:
```
a = df.groupby("city")["bedrooms"].count()
a
```

Out[33]:
```
city
Bangkok          17219
Cape Town        17707
Hong Kong         5857
Istanbul         22485
Mexico City      19256
New York         33404
Paris            51286
Rio de Janeiro   24869
Rome             26773
Sydney           31421
Name: bedrooms, dtype: int64
```

In [34]:
```
b = df.groupby("city")["review_scores_rating"].mean()
b
```

Out[34]:
```
city
Bangkok           93.001699
Cape Town         94.404838
Hong Kong         89.707517
Istanbul          91.063496
Mexico City       94.837959
New York          93.767188
Paris             93.063931
Rio de Janeiro    94.571349
Rome              93.516489
Sydney            93.234135
Name: review_scores_rating, dtype: float64
```

In [36]:
```
c = df.groupby("city")["review_scores_cleanliness"].mean()
c
```

Out[36]:
```
city
Bangkok           9.412901
Cape Town         9.530781
Hong Kong         8.992324
Istanbul          9.054278
Mexico City       9.564676
New York          9.268009
Paris             9.206446
Rio de Janeiro    9.392376
Rome              9.496687
Sydney            9.206995
Name: review_scores_cleanliness, dtype: float64
```

In [37]:
```
d = df.groupby("city")["price"].mean()
d
```

Out[37]:
```
city
Bangkok           2078.278033
Cape Town         2405.120350
Hong Kong          746.169889
Istanbul           532.557445
Mexico City       1149.253028
New York           142.842240
Paris              113.096445
Rio de Janeiro     742.589254
Rome               105.107643
Sydney             222.013440
Name: price, dtype: float64
```

conclusion

Bedrooms, review scores for rating and cleanliness significantaly influence the price of airbnb listings. Paris has the most bedrooms with high ratings and relatively low avg price of 113. In contrast, Rome has the lowest avg price at 105.

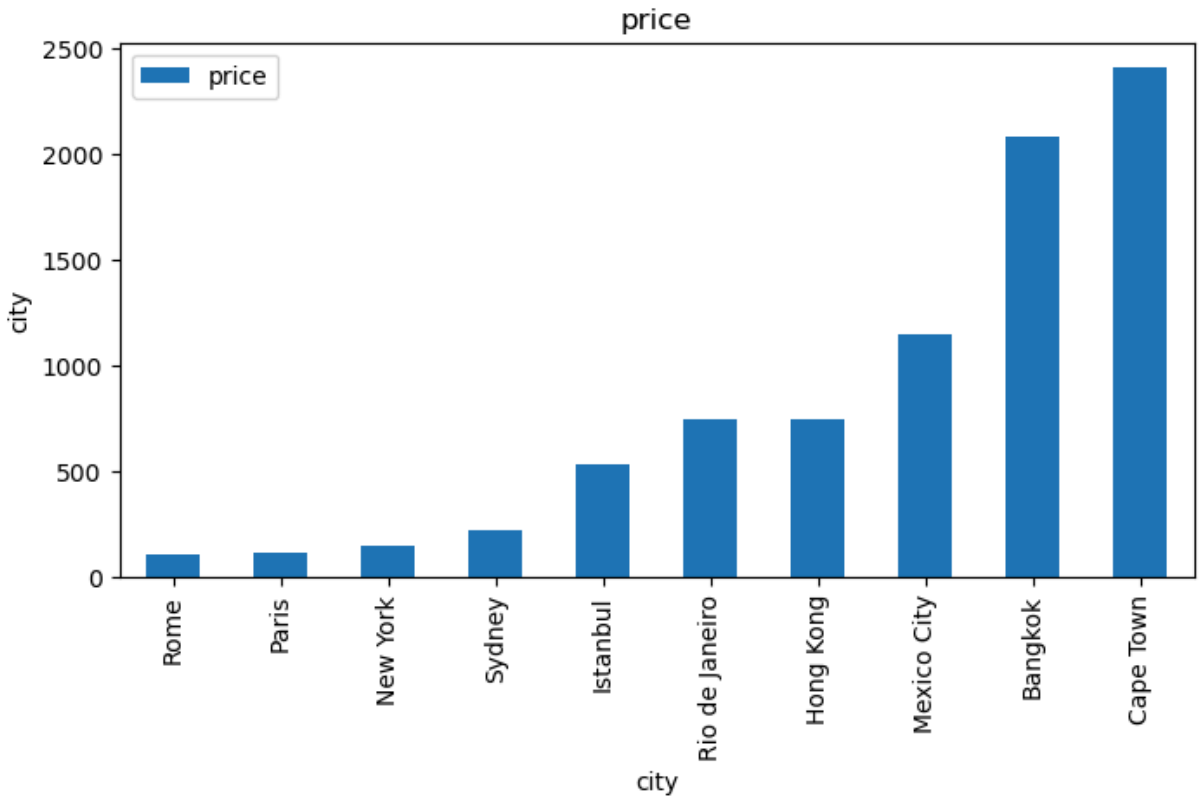
3. Which city offrers a better value of traval?

In [38]:
```
better_value =df.groupby('city')['price'].mean().sort_values(ascending = True)
better_value
```

Out[38]:
```
city
Rome               105.107643
Paris              113.096445
New York           142.842240
Sydney             222.013440
Istanbul           532.557445
Rio de Janeiro     742.589254
Hong Kong          746.169889
Mexico City       1149.253028
Bangkok           2078.278033
Cape Town         2405.120350
Name: price, dtype: float64
```

In [40]:
```
better_value.head(30).to_frame().plot(kind='bar',figsize=(8,4))
plt.ylabel('city')
plt.title('price')
```

Out[40]:
```
Text(0.5, 1.0, 'price')
```



conclusion

Rome offers the best traval value qwith the lowest price at 105 and excellent ratings.

4. Are you able to identify any trends or reasonality in the review data?

In [48]:
```
df2 = pd.read_csv("D:\\aman_new\\Reviews.csv\\Reviews.csv",encoding='latin1')
df2.head()
```

Out[48]:

	listing_id	review_id	date	reviewer_id
0	11798	330265172	2018-09-30	11863072
1	15383	330103585	2018-09-30	39147453
2	16455	329985788	2018-09-30	1125378
3	17919	330016899	2018-09-30	172717984
4	26827	329995638	2018-09-30	17542859

In [49]:
```
merge_df = df.merge(df2, how='inner', on=['listing_id'])
merge_df.head()
```

Out[49]:

	listing_id	name	host_id	host_since	host_location	host_response_time	host_res
0	281420	Beautiful Flat in le Village Montmartre, Paris	1466919	2011-12-03	Paris, Ile-de-France, France	NaN	
1	281420	Beautiful Flat in le Village Montmartre, Paris	1466919	2011-12-03	Paris, Ile-de-France, France	NaN	
2	3705183	39 mÃ□Â² Paris (Sacre CÃ□â□□ur)	10328771	2013-11-29	Paris, Ile-de-France, France	NaN	
3	3705183	39 mÃ□Â² Paris (Sacre CÃ□â□□ur)	10328771	2013-11-29	Paris, Ile-de-France, France	NaN	
4	3705183	39 mÃ□Â² Paris (Sacre CÃ□â□□ur)	10328771	2013-11-29	Paris, Ile-de-France, France	NaN	

5 rows × 36 columns

In [51]:
```
merge_df['Months'] = pd.to_datetime(merge_df['date'], infer_datetime_format=True).d
merge_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5373143 entries, 0 to 5373142
Data columns (total 37 columns):
 #   Column                       Dtype  
---  ------                       -----  
 0   listing_id                   int64  
 1   name                         object 
 2   host_id                      int64  
 3   host_since                   object 
 4   host_location                object 
 5   host_response_time           object 
 6   host_response_rate           float64
 7   host_acceptance_rate         float64
 8   host_is_superhost            object 
 9   host_total_listings_count    float64
 10  host_has_profile_pic         object 
 11  host_identity_verified       object 
 12  neighbourhood                object 
 13  district                     object 
 14  city                         object 
 15  latitude                     float64
 16  longitude                    float64
 17  property_type                object 
 18  room_type                    object 
 19  accommodates                 int64  
 20  bedrooms                     float64
 21  amenities                    object 
 22  price                        int64  
 23  minimum_nights               int64  
 24  maximum_nights               int64  
 25  review_scores_rating         float64
 26  review_scores_accuracy       float64
 27  review_scores_cleanliness    float64
 28  review_scores_checkin        float64
 29  review_scores_communication  float64
 30  review_scores_location       float64
 31  review_scores_value          float64
 32  instant_bookable             object 
 33  review_id                    int64  
 34  date                         object 
 35  reviewer_id                  int64  
 36  Months                       int32  
dtypes: float64(13), int32(1), int64(8), object(15)
memory usage: 1.5+ GB
```

In [53]:
```
merge_df['date'].value_counts().sort_values(ascending=False)
```

Out[53]:
```
date
2020-01-02    10136
2020-01-01     9635
2019-10-06     9423
2019-11-03     8937
2019-09-29     8905
               ...
2009-12-29        1
2011-01-20        1
2010-05-10        1
2010-05-19        1
2010-06-03        1
Name: count, Length: 4103, dtype: int64
```

In [54]:
```
merge_df.groupby('Months').agg('Months').count().to_frame().plot(kind='bar',figsize

plt.ylabel('No. of pizza ordered per month')
plt.title('No. of pizza sold per month')
```

Out[54]:
```
Text(0.5, 1.0, 'No. of pizza sold per month')
```

conclusion

The higgest bookings occur in the auntumn months of september and october, and in winter during january. The record for the most bookings in a single day is 10,136 rooms on jaunuary 2, 2020

In [ ]: