PREVISÃO DE PREÇOS LOCAÇÃO AIRBNB

In [1]:
```
import pandas as pd
import pathlib
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor, ExtraTreesRegressor
from sklearn.model_selection import train_test_split
```

In [2]:
```
data = pd.read_csv(r'C:\Users\RENILDO BONIFÁCIO\PROJETO MACHINE LEARNING REGRESSAO\janeiro2020.csv')
data.head()
```

```
C:\Anaconda\lib\site-packages\IPython\core\interactiveshell.py:3165: DtypeWarning: Columns (61,62,94) have mixed types.Specify dty
pe option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

Out[2]:

	id	listing_url	scrape_id	last_scraped	name	summary	space	description	experiences_offered	nei
0	17878	https://www.airbnb.com/rooms/17878	20200121213543	2020-01-22	Very Nice 2Br - Copacabana - WiFi	Pls note that special rates apply for Carnival...	- large balcony which looks out on pedestrian ...	Pls note that special rates apply for Carnival...	none	F
1	21280	https://www.airbnb.com/rooms/21280	20200121213543	2020-01-22	Renovated Modern Apt. Near Beach	Immaculately renovated top-floor apartment ove...	Immaculately renovated top-floor apartment in ...	Immaculately renovated top-floor apartment ove...	none	
2	25026	https://www.airbnb.com/rooms/25026	20200121213543	2020-01-22	Beautiful Modern Decorated Studio in Copa	Our apartment is a little gem, everyone loves ...	This newly renovated studio (last renovations ...	Our apartment is a little gem, everyone loves ...	none	

	id	listing_url	scrape_id	last_scraped	name	summary	space	description	experiences_offered	nei
3	31560	https://www.airbnb.com/rooms/31560	20200121213543	2020-01-22	NICE & COZY 1BDR - IPANEMA BEACH	This nice and clean 1 bedroom apartment is loc...	This nice and clean 1 bedroom apartment is loc...	This nice and clean 1 bedroom apartment is loc...	none	
4	35636	https://www.airbnb.com/rooms/35636	20200121213543	2020-01-22	Cosy flat close to Ipanema beach	This cosy apartment is just a few steps away ...	The location is extremely convenient, safe and...	This cosy apartment is just a few steps away ...	none	-

5 rows × 106 columns

In [3]:
```
print(list(data.columns))
```

```
['id', 'listing_url', 'scrape_id', 'last_scraped', 'name', 'summary', 'space', 'description', 'experiences_offered', 'neighborhood
_overview', 'notes', 'transit', 'access', 'interaction', 'house_rules', 'thumbnail_url', 'medium_url', 'picture_url', 'xl_picture_
url', 'host_id', 'host_url', 'host_name', 'host_since', 'host_location', 'host_about', 'host_response_time', 'host_response_rate',
'host_acceptance_rate', 'host_is_superhost', 'host_thumbnail_url', 'host_picture_url', 'host_neighbourhood', 'host_listings_coun
t', 'host_total_listings_count', 'host_verifications', 'host_has_profile_pic', 'host_identity_verified', 'street', 'neighbourhoo
d', 'neighbourhood_cleansed', 'neighbourhood_group_cleansed', 'city', 'state', 'zipcode', 'market', 'smart_location', 'country_cod
e', 'country', 'latitude', 'longitude', 'is_location_exact', 'property_type', 'room_type', 'accommodates', 'bathrooms', 'bedroom
s', 'beds', 'bed_type', 'amenities', 'square_feet', 'price', 'weekly_price', 'monthly_price', 'security_deposit', 'cleaning_fee',
'guests_included', 'extra_people', 'minimum_nights', 'maximum_nights', 'minimum_minimum_nights', 'maximum_minimum_nights', 'minimu
m_maximum_nights', 'maximum_maximum_nights', 'minimum_nights_avg_ntm', 'maximum_nights_avg_ntm', 'calendar_updated', 'has_availabi
lity', 'availability_30', 'availability_60', 'availability_90', 'availability_365', 'calendar_last_scraped', 'number_of_reviews',
'number_of_reviews_ltm', 'first_review', 'last_review', 'review_scores_rating', 'review_scores_accuracy', 'review_scores_cleanline
ss', 'review_scores_checkin', 'review_scores_communication', 'review_scores_location', 'review_scores_value', 'requires_license',
'license', 'jurisdiction_names', 'instant_bookable', 'is_business_travel_ready', 'cancellation_policy', 'require_guest_profile_pic
ture', 'require_guest_phone_verification', 'calculated_host_listings_count', 'calculated_host_listings_count_entire_homes', 'calcu
lated_host_listings_count_private_rooms', 'calculated_host_listings_count_shared_rooms', 'reviews_per_month']
```

In [4]:
```
print(data.isnull().sum())
```

```
id                                              0
listing_url                                     0
scrape_id                                       0
last_scraped                                    0
name                                           58
                                              ...
```

```
calculated_host_listings_count                    0
calculated_host_listings_count_entire_homes       0
calculated_host_listings_count_private_rooms      0
calculated_host_listings_count_shared_rooms       0
reviews_per_month                             14406
Length: 106, dtype: int64
```

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

Out[5]:

	id	scrape_id	thumbnail_url	medium_url	xl_picture_url	host_id	host_acceptance_rate	host_listings_count	host_total_listings_c
count	3.475400e+04	3.475400e+04	0.0	0.0	0.0	3.475400e+04	0.0	34749.000000	34749.00(
mean	1.993875e+07	2.020012e+13	NaN	NaN	NaN	7.856555e+07	NaN	6.992518	6.99
std	1.284913e+07	1.266815e+01	NaN	NaN	NaN	8.242840e+07	NaN	29.215266	29.21
min	1.787800e+04	2.020012e+13	NaN	NaN	NaN	1.173900e+04	NaN	0.000000	0.00(
25%	1.069316e+07	2.020012e+13	NaN	NaN	NaN	1.464975e+07	NaN	1.000000	1.00(
50%	1.589669e+07	2.020012e+13	NaN	NaN	NaN	5.447367e+07	NaN	1.000000	1.00(
75%	3.185959e+07	2.020012e+13	NaN	NaN	NaN	1.003698e+08	NaN	3.000000	3.00(
max	4.174458e+07	2.020012e+13	NaN	NaN	NaN	3.294585e+08	NaN	1321.000000	1321.00(

8 rows × 45 columns

In [6]:
```
colunas = ['host_listings_count','latitude','longitude','property_type','room_type','accommodates','bathrooms','bedrooms','beds','

data = data.loc[:, colunas]
print(list(data.columns))
data.head()
```

```
['host_listings_count', 'latitude', 'longitude', 'property_type', 'room_type', 'accommodates', 'bathrooms', 'bedrooms', 'beds', 'b
ed_type', 'price', 'cleaning_fee', 'guests_included', 'extra_people', 'minimum_nights', 'maximum_nights', 'number_of_reviews']
```

Out[6]:

	host_listings_count	latitude	longitude	property_type	room_type	accommodates	bathrooms	bedrooms	beds	bed_type	price	cleaning_fee	gue
0	2.0	-22.96592	-43.17896	Condominium	Entire home/apt	5	1.0	2.0	2.0	Real Bed	$332.00	$378.00	

	host_listings_count	latitude	longitude	property_type	room_type	accommodates	bathrooms	bedrooms	beds	bed_type	price	cleaning_fee	gue
1	0.0	-22.98467	-43.19611	Apartment	Entire home/apt	6	2.0	2.0	4.0	Real Bed	$336.00	$210.00	
2	3.0	-22.97712	-43.19045	Apartment	Entire home/apt	2	1.0	1.0	2.0	Real Bed	$159.00	$250.00	
3	1.0	-22.98302	-43.21427	Apartment	Entire home/apt	3	1.0	1.0	2.0	Real Bed	$273.00	$84.00	
4	1.0	-22.98816	-43.19359	Apartment	Entire home/apt	2	1.5	1.0	1.0	Real Bed	$378.00	$172.00	

In [7]:
```
print(data.isnull().sum())
```

```
host_listings_count        5
latitude                   0
longitude                  0
property_type              0
room_type                  0
accommodates               0
bathrooms                 54
bedrooms                  60
beds                     175
bed_type                   0
price                      0
cleaning_fee           10914
guests_included            0
extra_people               0
minimum_nights             0
maximum_nights             0
number_of_reviews          0
dtype: int64
```

In [8]:
```
data = data.dropna()
print(data.shape)
print(data.isnull().sum())
```

```
(23741, 17)
host_listings_count    0
latitude               0
longitude              0
```

```
property_type          0
room_type              0
accommodates           0
bathrooms              0
bedrooms               0
beds                   0
bed_type               0
price                  0
cleaning_fee           0
guests_included        0
extra_people           0
minimum_nights         0
maximum_nights         0
number_of_reviews      0
dtype: int64
```

In [9]:

```
print(data.dtypes)
print('-'*60)
print(data.iloc[0])
```

```
host_listings_count    float64
latitude               float64
longitude              float64
property_type           object
room_type               object
accommodates             int64
bathrooms              float64
bedrooms               float64
beds                   float64
bed_type                object
price                   object
cleaning_fee            object
guests_included          int64
extra_people            object
minimum_nights           int64
maximum_nights           int64
number_of_reviews        int64
dtype: object
------------------------------------------------------------
host_listings_count               2.0
latitude                    -22.96592
longitude                   -43.17896
property_type             Condominium
room_type             Entire home/apt
accommodates                        5
bathrooms                         1.0
```

```
bedrooms                          2.0
beds                              2.0
bed_type                     Real Bed
price                         $332.00
cleaning_fee                  $378.00
guests_included                     2
extra_people                 $63.00
minimum_nights                      5
maximum_nights                     30
number_of_reviews                 246
Name: 0, dtype: object
```

In [10]:

```
# modificar tipos das colunas (price, cleaning_fee, extra_people)

data['price'] = data['price'].str.replace('$', '').str.replace(',', '').astype(np.float64, copy=False)
data['cleaning_fee'] = data['cleaning_fee'].str.replace('$', '').str.replace(',', '').astype(np.float64, copy=False)
data['extra_people'] = data['extra_people'].str.replace('$', '').str.replace(',', '').astype(np.float64, copy=False)

print(data.dtypes)
```

```
host_listings_count    float64
latitude               float64
longitude              float64
property_type           object
room_type               object
accommodates             int64
bathrooms              float64
bedrooms               float64
beds                   float64
bed_type                object
price                  float64
cleaning_fee           float64
guests_included          int64
extra_people           float64
minimum_nights           int64
maximum_nights           int64
number_of_reviews        int64
dtype: object
<ipython-input-10-46770d67877f>:3: FutureWarning: The default value of regex will change from True to False in a future version. I
n addition, single character regular expressions will*not* be treated as literal strings when regex=True.
  data['price'] = data['price'].str.replace('$', '').str.replace(',', '').astype(np.float64, copy=False)
<ipython-input-10-46770d67877f>:4: FutureWarning: The default value of regex will change from True to False in a future version. I
n addition, single character regular expressions will*not* be treated as literal strings when regex=True.
  data['cleaning_fee'] = data['cleaning_fee'].str.replace('$', '').str.replace(',', '').astype(np.float64, copy=False)
<ipython-input-10-46770d67877f>:5: FutureWarning: The default value of regex will change from True to False in a future version. I
```

```
n addition, single character regular expressions will*not* be treated as literal strings when regex=True.
  data['extra_people'] = data['extra_people'].str.replace('$', '').str.replace(',', '').astype(np.float64, copy=False)
```

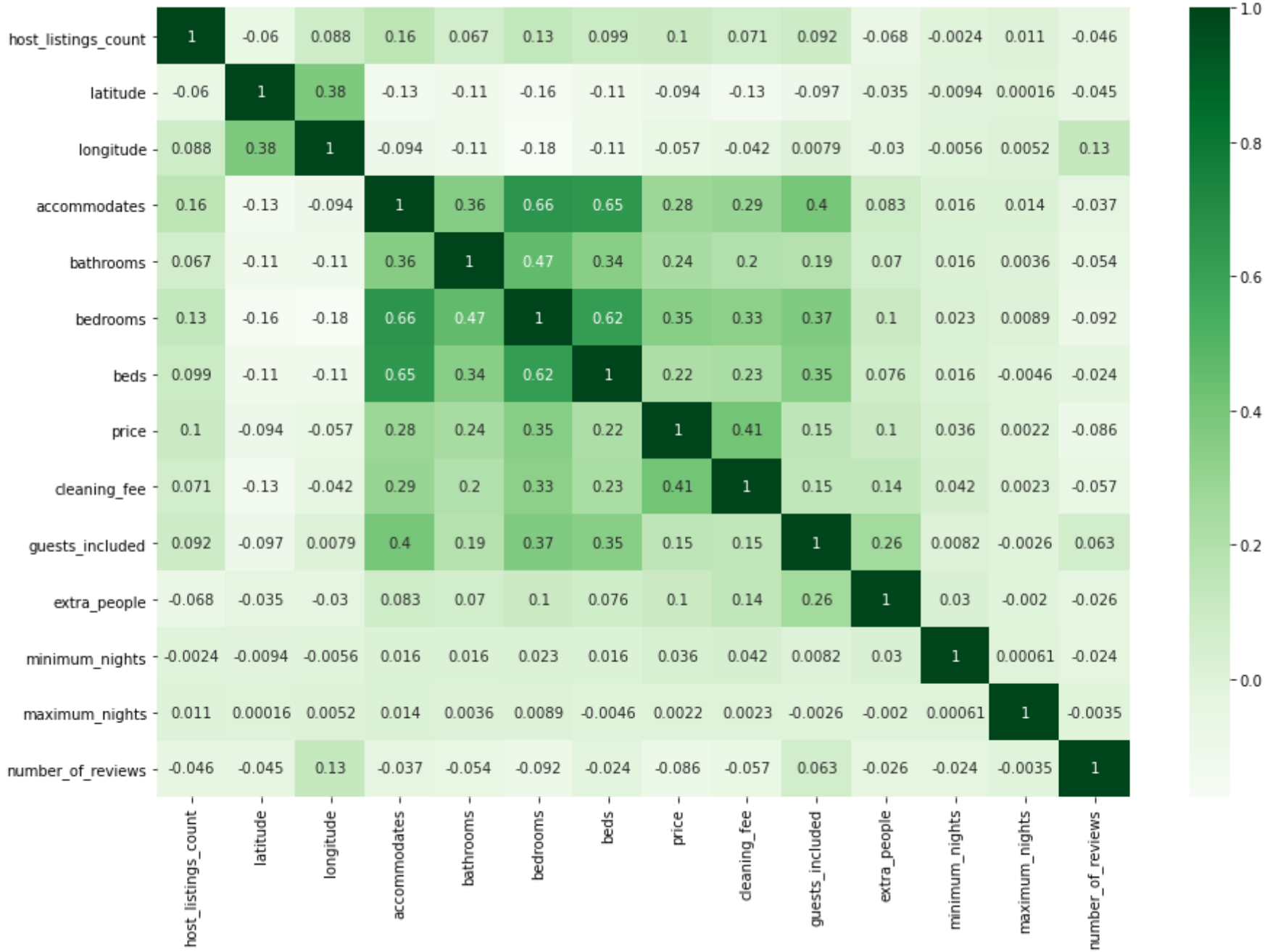
In [11]:
```
data.head()
```

Out[11]:

	host_listings_count	latitude	longitude	property_type	room_type	accommodates	bathrooms	bedrooms	beds	bed_type	price	cleaning_fee	guest
0	2.0	-22.96592	-43.17896	Condominium	Entire home/apt	5	1.0	2.0	2.0	Real Bed	332.0	378.0	
1	0.0	-22.98467	-43.19611	Apartment	Entire home/apt	6	2.0	2.0	4.0	Real Bed	336.0	210.0	
2	3.0	-22.97712	-43.19045	Apartment	Entire home/apt	2	1.0	1.0	2.0	Real Bed	159.0	250.0	
3	1.0	-22.98302	-43.21427	Apartment	Entire home/apt	3	1.0	1.0	2.0	Real Bed	273.0	84.0	
4	1.0	-22.98816	-43.19359	Apartment	Entire home/apt	2	1.5	1.0	1.0	Real Bed	378.0	172.0	

In [12]:
```
plt.figure(figsize=(15, 10))
sns.heatmap(data.corr(), annot=True, cmap='Greens')
#print(data.corr())
```

Out[12]:
```
<AxesSubplot:>
```



ANÁLISE DAS PROPRIEDADES DE CADA GRUPO

In [13]:
```
data['property_type'].value_counts()

# Analisando os tipos de propriedades que tem valores pequenos, uma opção é agrupar em um único grupo
```

Out[13]:
```
Apartment                18566
House                     1848
Condominium               1540
Loft                       542
Serviced apartment         488
Guest suite                136
Guesthouse                  99
Hostel                      74
Bed and breakfast           66
Villa                       66
Townhouse                   58
Aparthotel                  42
Hotel                       41
Other                       31
Cottage                     26
Chalet                      22
Boutique hotel              19
Tiny house                  18
Earth house                 12
Nature lodge                 9
Bungalow                     7
Boat                         7
Cabin                        6
Island                       5
Houseboat                    2
Treehouse                    2
Casa particular (Cuba)       2
Farm stay                    2
Tent                         1
Camper/RV                    1
Campsite                     1
Hut                          1
Yurt                         1
Name: property_type, dtype: int64
```

In [14]:
```
data['beds'].value_counts()
```

Out[14]:
```
1.0     6973
2.0     6718
3.0     4343
```

```
4.0     2524
5.0     1196
6.0      766
0.0      382
7.0      277
8.0      227
10.0      92
9.0       81
12.0      41
11.0      27
16.0      25
13.0      14
14.0      14
15.0      10
18.0       6
17.0       5
20.0       3
35.0       2
38.0       2
30.0       2
21.0       1
62.0       1
29.0       1
33.0       1
26.0       1
44.0       1
27.0       1
22.0       1
39.0       1
69.0       1
50.0       1
Name: beds, dtype: int64
```

In [15]:
```
data['bedrooms'].value_counts()
```

Out[15]:
```
1.0     11677
2.0      6447
3.0      3147
0.0      1393
4.0       739
5.0       202
6.0        84
7.0        26
11.0        7
8.0         6
9.0         5
```

```
20.0        2
10.0        2
12.0        2
15.0        1
17.0        1
Name: bedrooms, dtype: int64
```

Definindos minhas funções para Análise de Outliers

Analisando as colunas numéricas

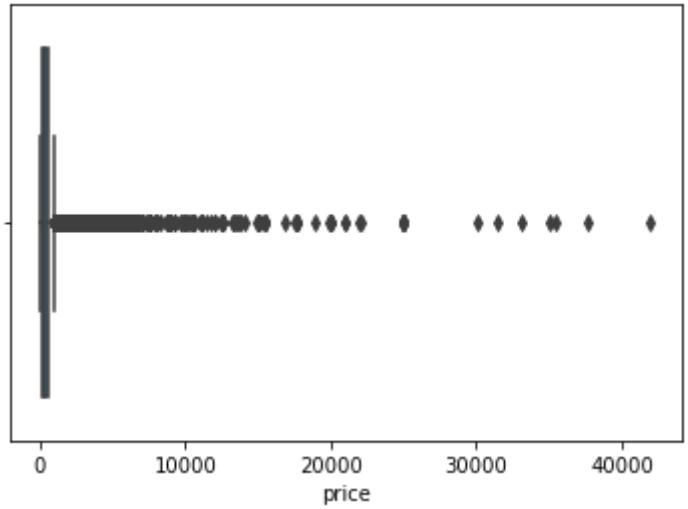
In [16]:
```
def diagrama_caixa(coluna):
    sns.boxplot(x=coluna)

def histograma(coluna):
    plt.figure(figsize=(15, 5))
    sns.distplot(coluna, hist=True)

def grafico_barra(coluna):
    plt.figure(figsize=(15, 5))
    plt.tick_params(axis='x', rotation=90)
    ax = sns.barplot(x=coluna.value_counts().index, y=coluna.value_counts())
```

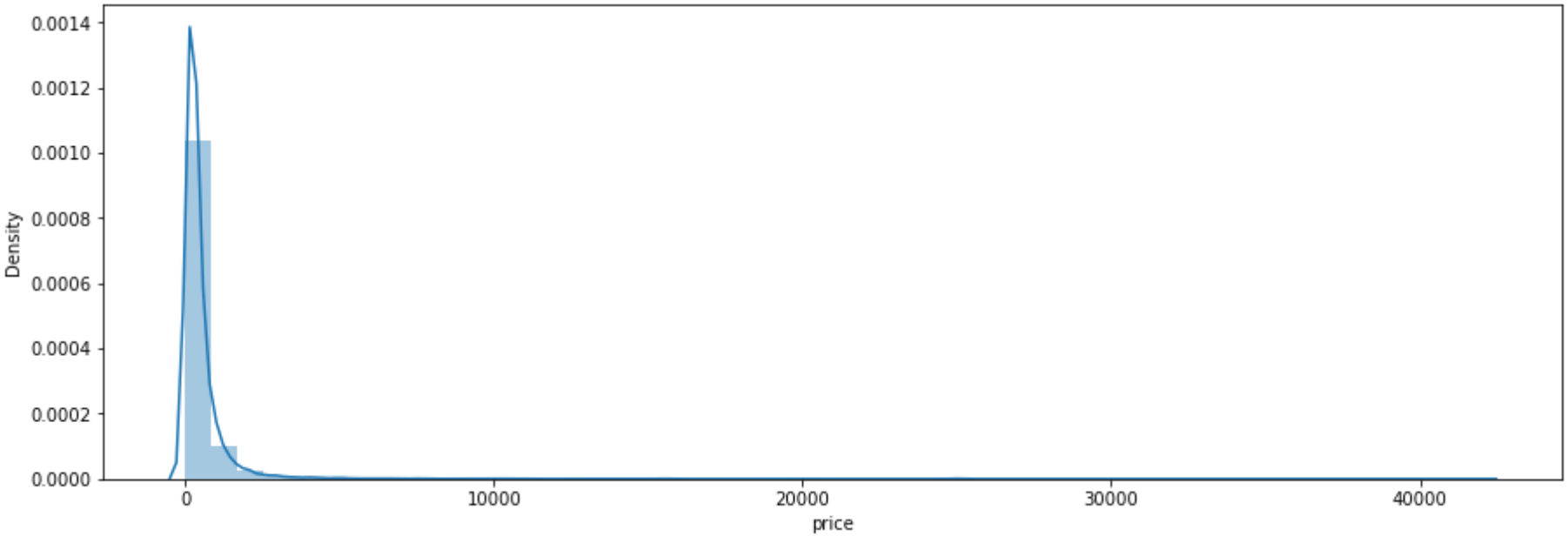
price

In [17]:
```
diagrama_caixa(data['price'])
histograma(data['price'])
data.shape
```

```
C:\Anaconda\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be remove
d in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histp
lot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[17]: (23741, 17)





In [18]:

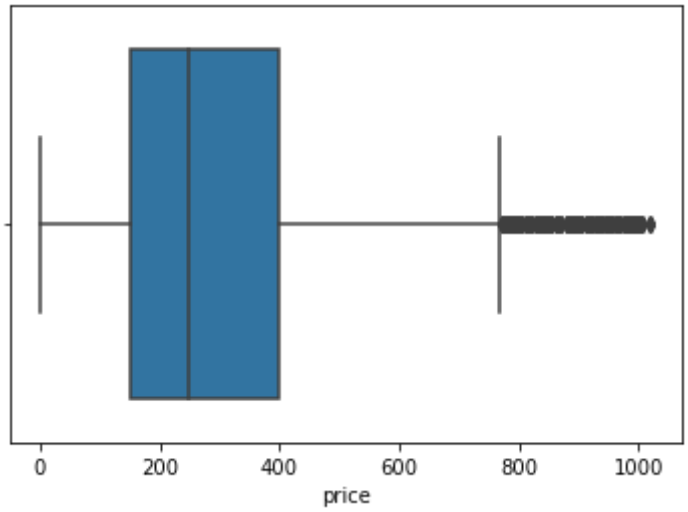
```
# EXCLUINDO OUTLIERS

Q1 = data['price'].quantile(.25)
Q3 = data['price'].quantile(.75)
IQR = Q3 - Q1
LimI = Q1 - (1.5 * IQR)
```

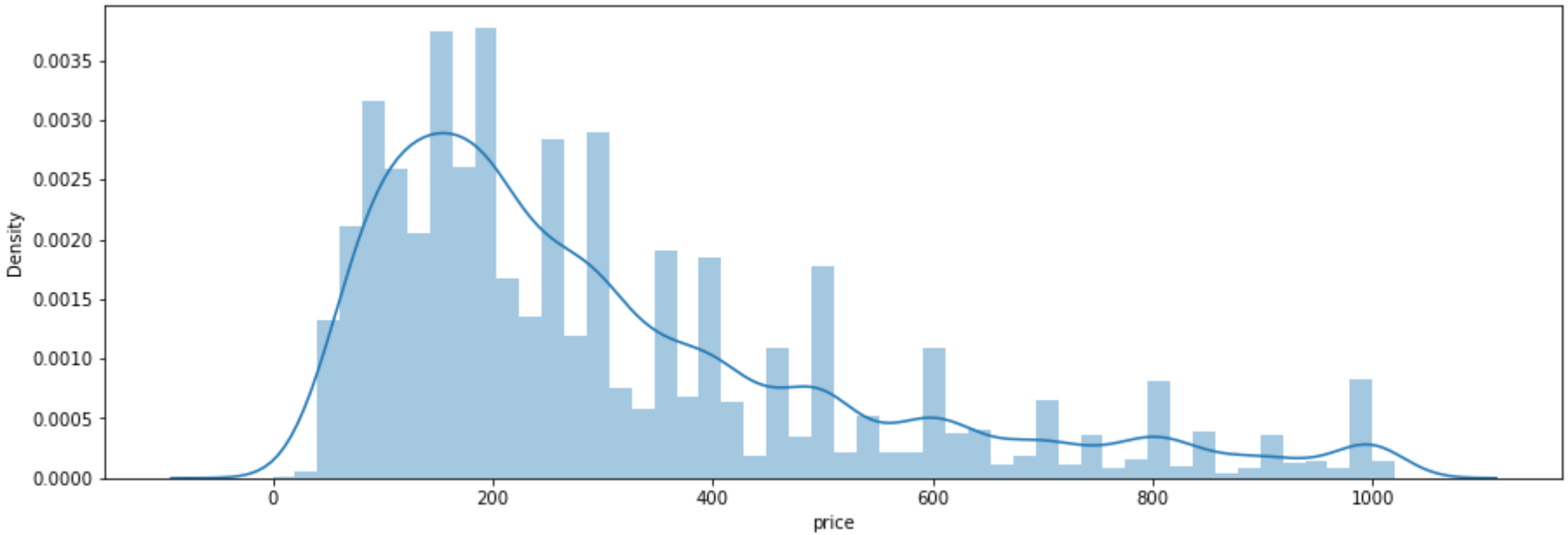
```
LimSP = Q3 + (1.5 * IQR)
data = data.loc[(data['price'] >= LimI) & (data['price'] <= LimSP)]
data.shape
```

Out[18]: (21559, 17)

In [19]:
```
diagrama_caixa(data['price'])
histograma(data['price'])
```

```
C:\Anaconda\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be remove
d in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histp
lot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



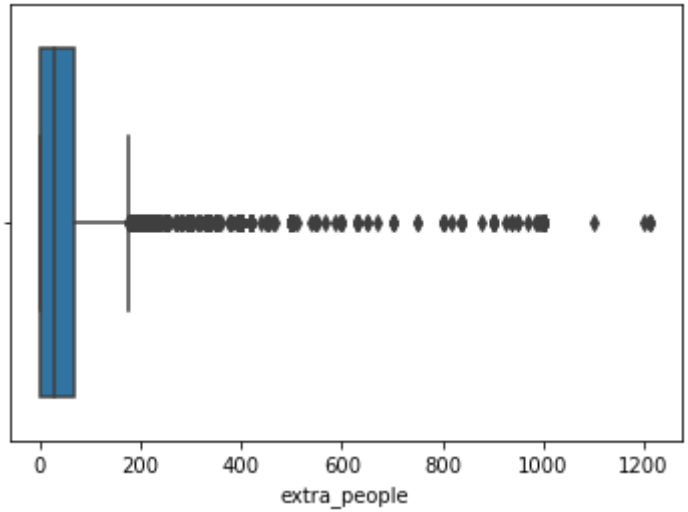


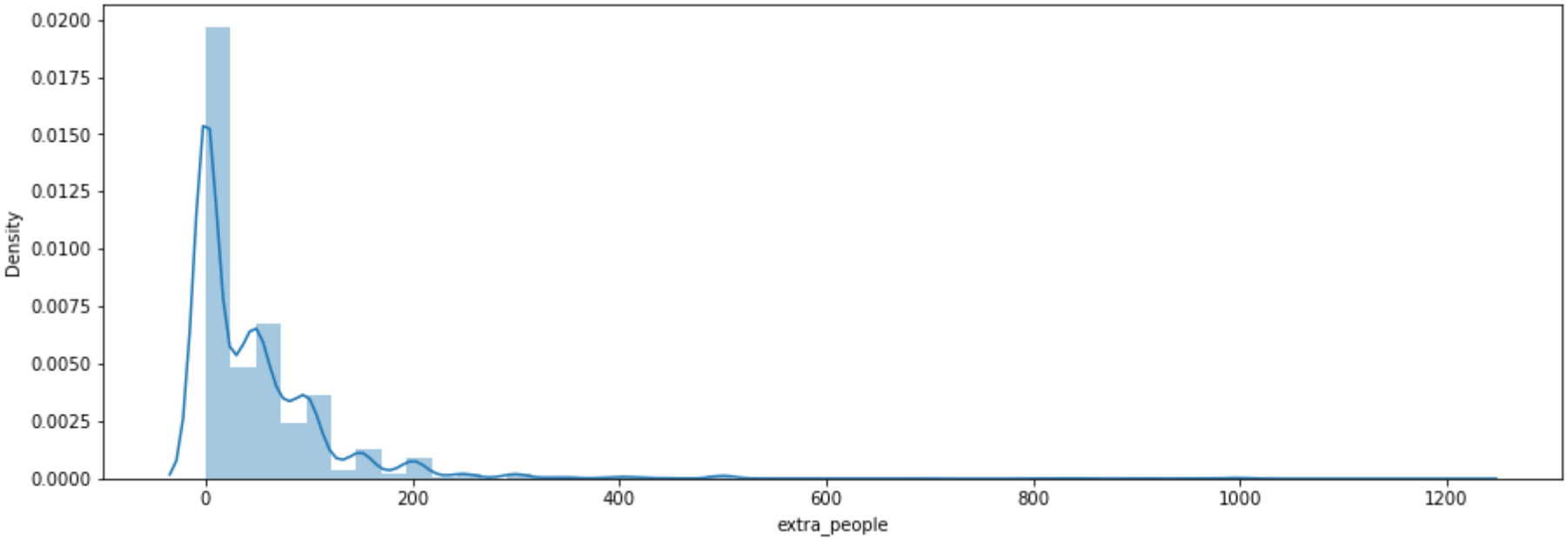
extra_people

In [20]:
```
diagrama_caixa(data['extra_people'])
histograma(data['extra_people'])
data.shape
```

```
C:\Anaconda\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be remove
d in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histp
lot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[20]: (21559, 17)





In [21]:

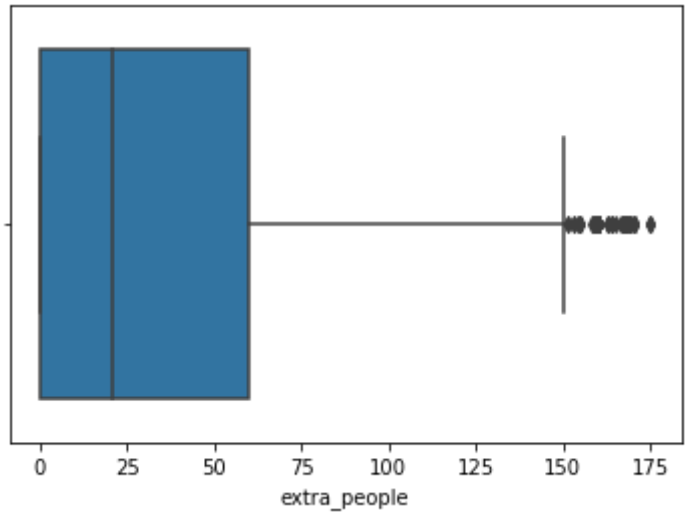
```
# EXCLUINDO OUTLIERS

Q1 = data['extra_people'].quantile(.25)
Q3 = data['extra_people'].quantile(.75)
IQR = Q3 - Q1
LimI = Q1 - (1.5 * IQR)
```

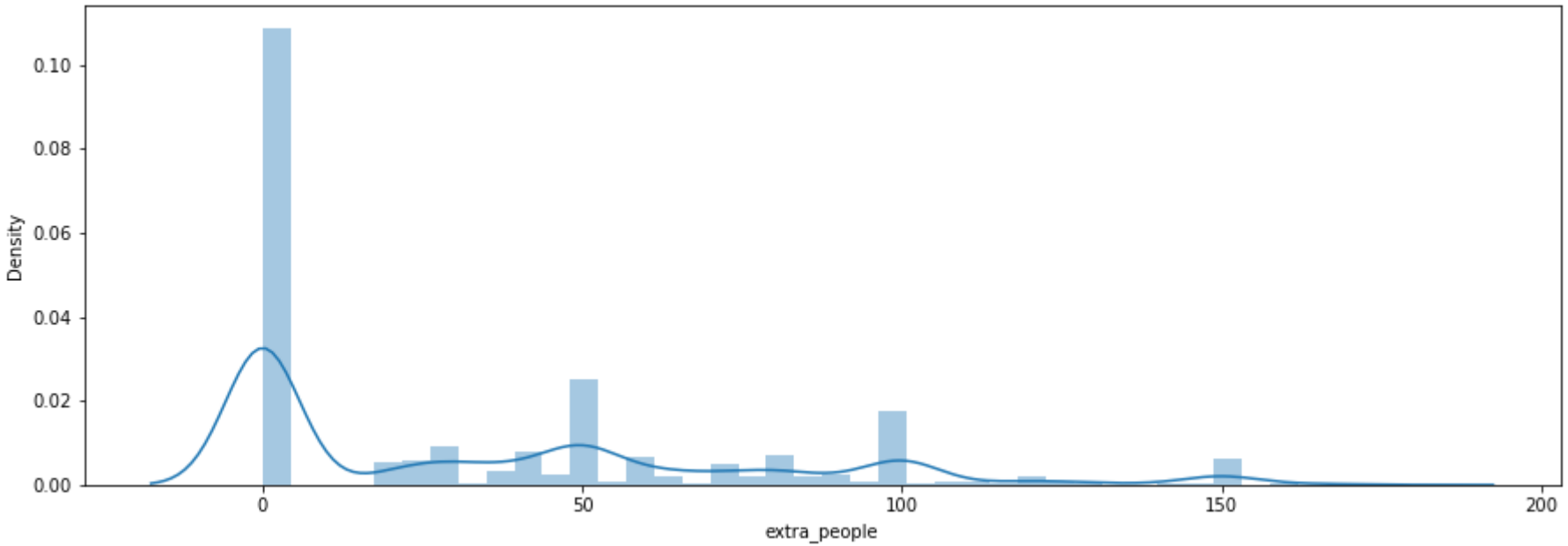
```
LimSP = Q3 + (1.5 * IQR)
data = data.loc[(data['extra_people'] >= LimI) & (data['extra_people'] <= LimSP)]
data.shape
```

Out[21]: (20411, 17)

In [22]:
```
diagrama_caixa(data['extra_people'])
histograma(data['extra_people'])
```

```
C:\Anaconda\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be remove
d in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histp
lot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

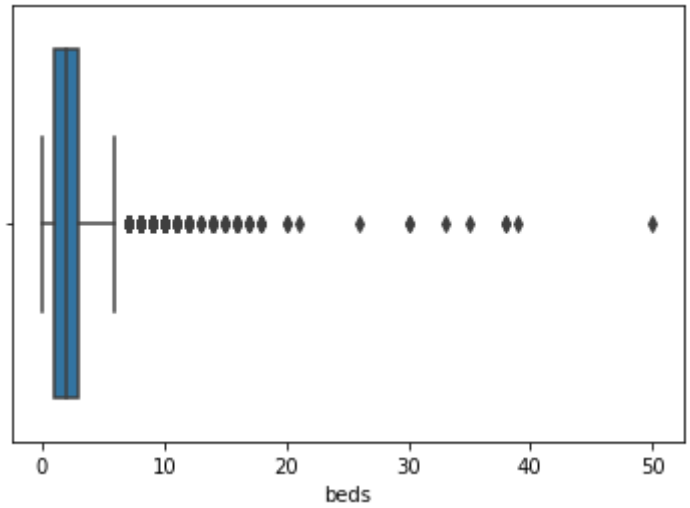


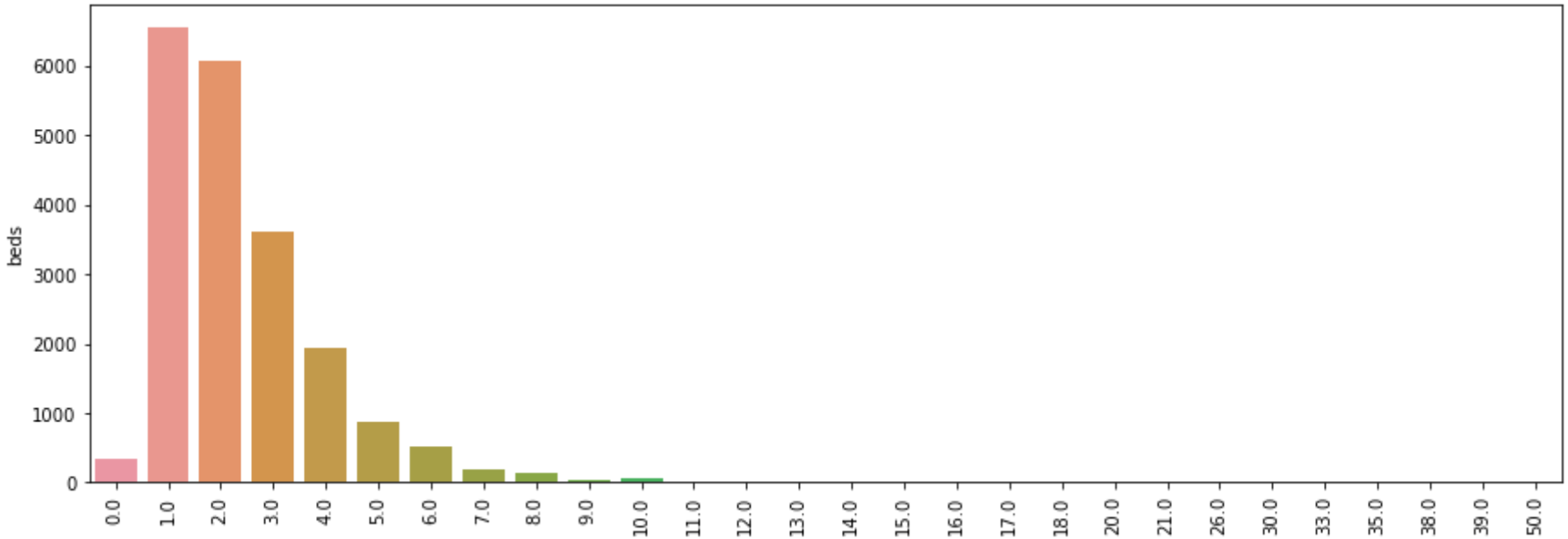


beds

In [23]:
```
diagrama_caixa(data['beds'])
grafico_barra(data['beds'])
data.shape
```

Out[23]: (20411, 17)





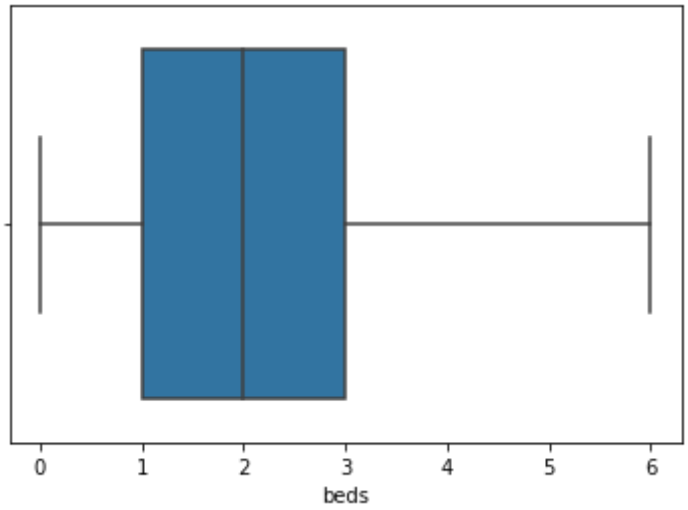
In [24]:

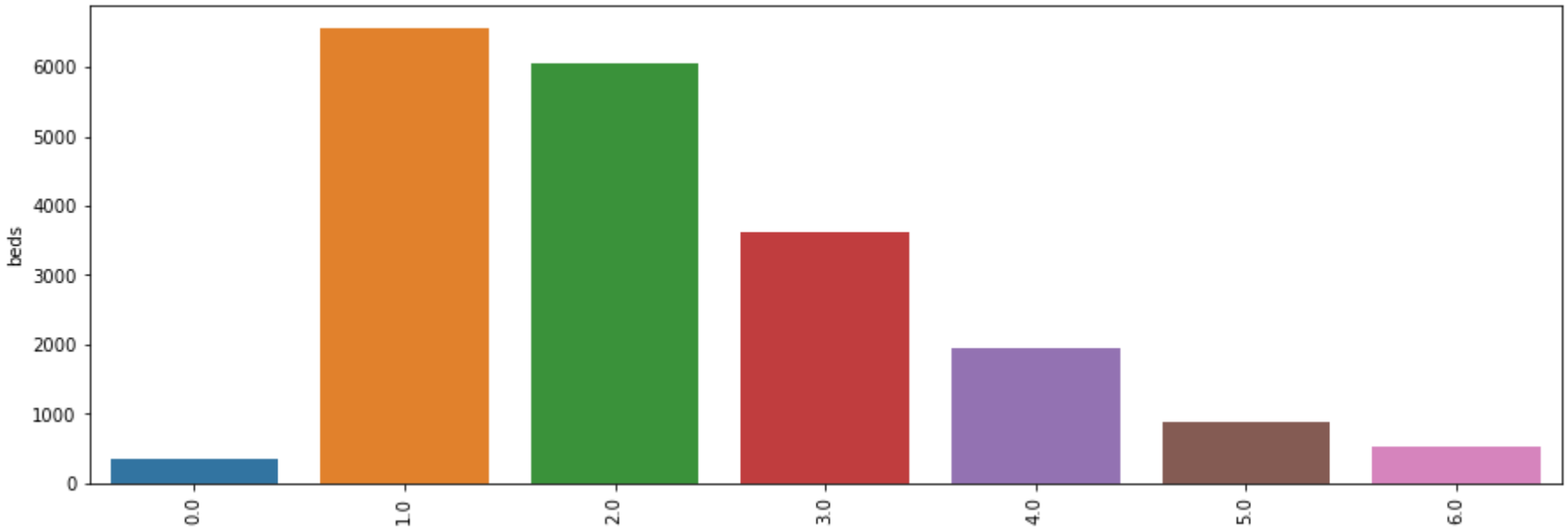
```
# EXCLUINDO OUTLIERS

Q1 = data['beds'].quantile(.25)
Q3 = data['beds'].quantile(.75)
IQR = Q3 - Q1
LimI = Q1 - (1.5 * IQR)
```

```
LimSP = Q3 + (1.5 * IQR)
data = data.loc[(data['beds'] >= LimI) & (data['beds'] <= LimSP)]
data.shape
```

Out[24]: (19914, 17)

In [25]:
```
diagrama_caixa(data['beds'])
grafico_barra(data['beds'])
```

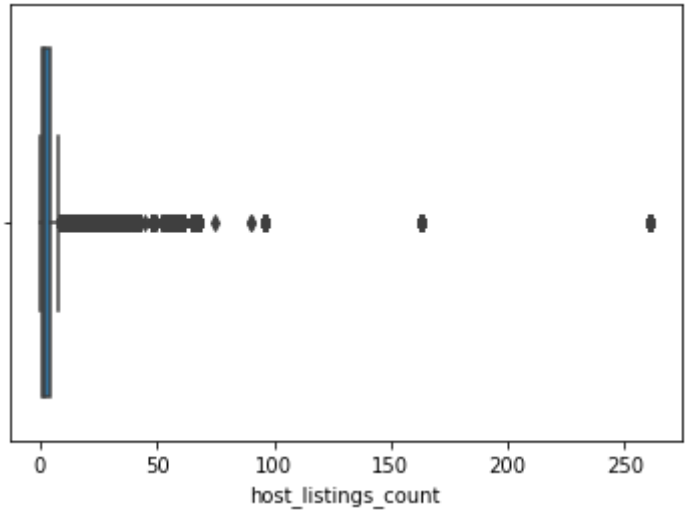


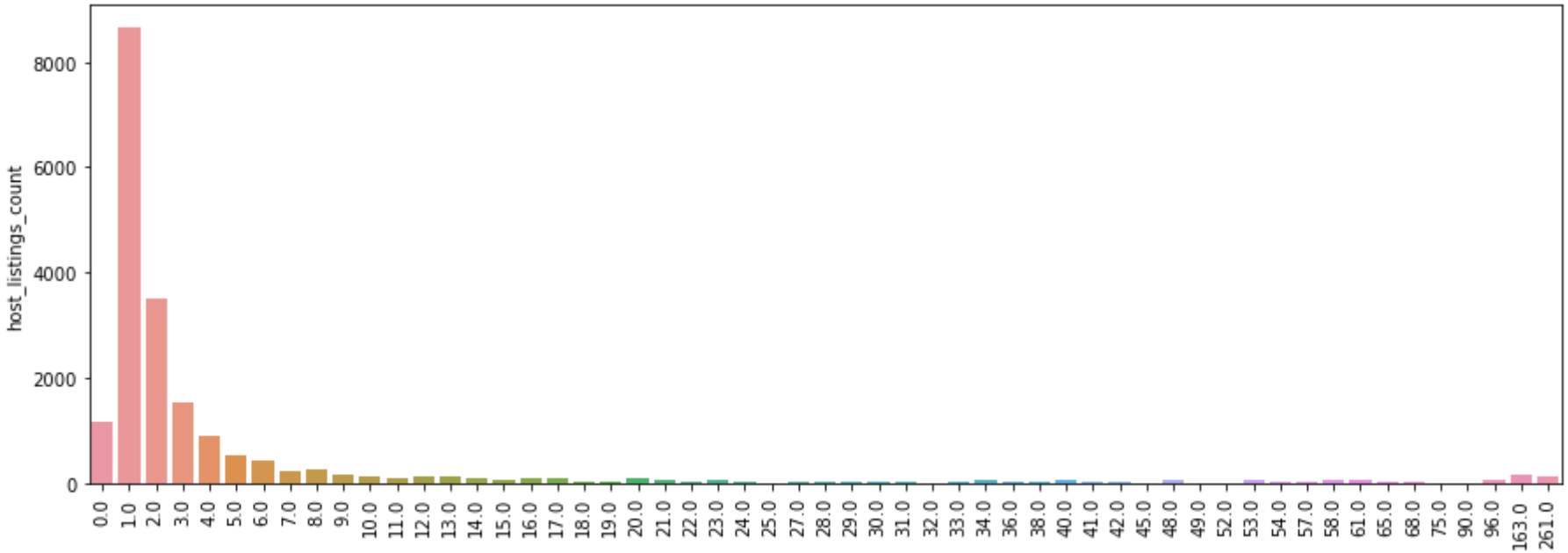


host_listing_count

In [26]:
```
diagrama_caixa(data['host_listings_count'])
grafico_barra(data['host_listings_count'])
data.shape
```

Out[26]: (19914, 17)





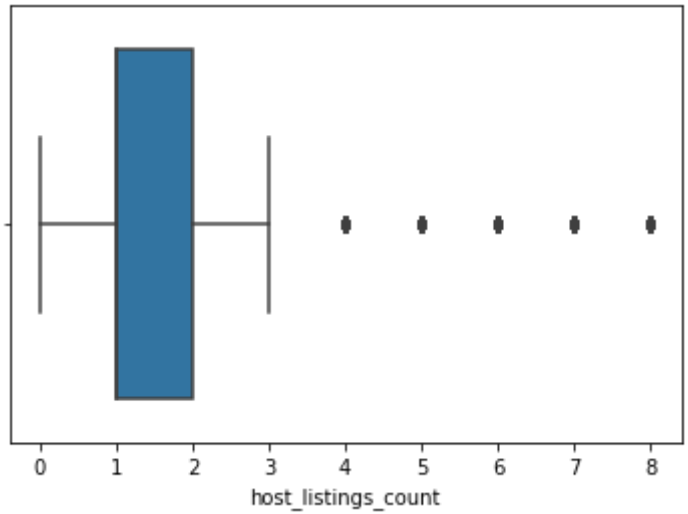
In [27]:

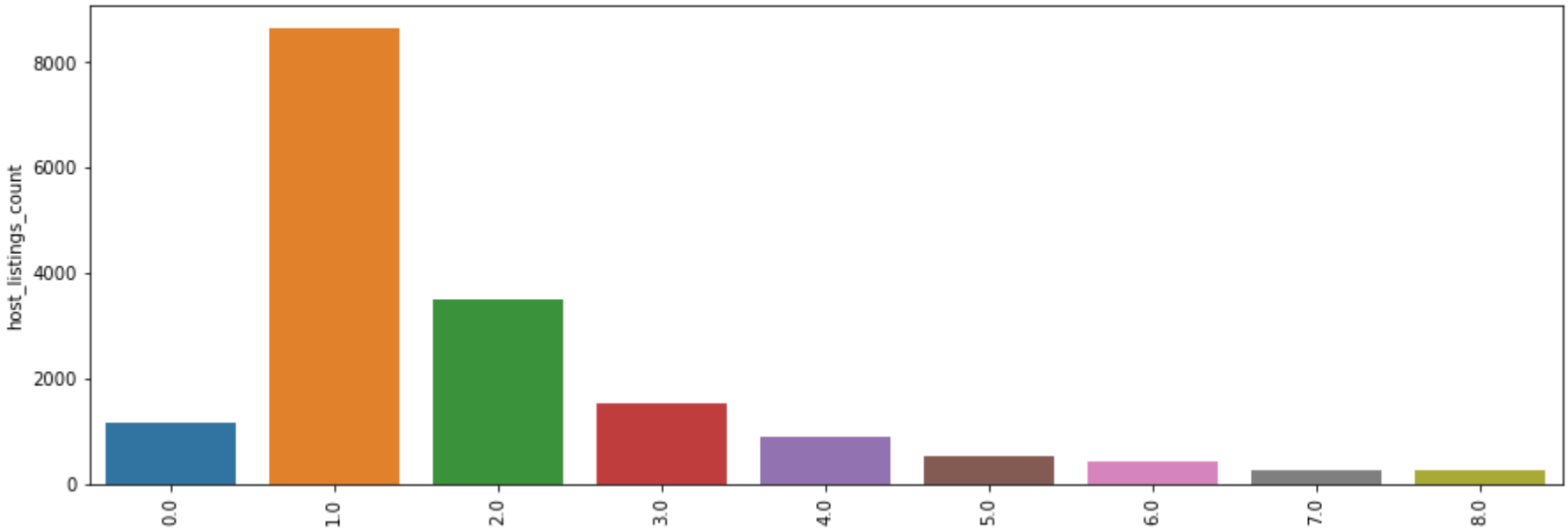
```
# EXCLUINDO OUTLIERS

Q1 = data['host_listings_count'].quantile(.25)
Q3 = data['host_listings_count'].quantile(.75)
IQR = Q3 - Q1
LimI = Q1 - (1.5 * IQR)
```

```
LimSP = Q3 + (1.5 * IQR)
data = data.loc[(data['host_listings_count'] >= LimI) & (data['host_listings_count'] <= LimSP)]
data.shape
```

Out[27]: (17232, 17)

In [28]:
```
diagrama_caixa(data['host_listings_count'])
grafico_barra(data['host_listings_count'])
```

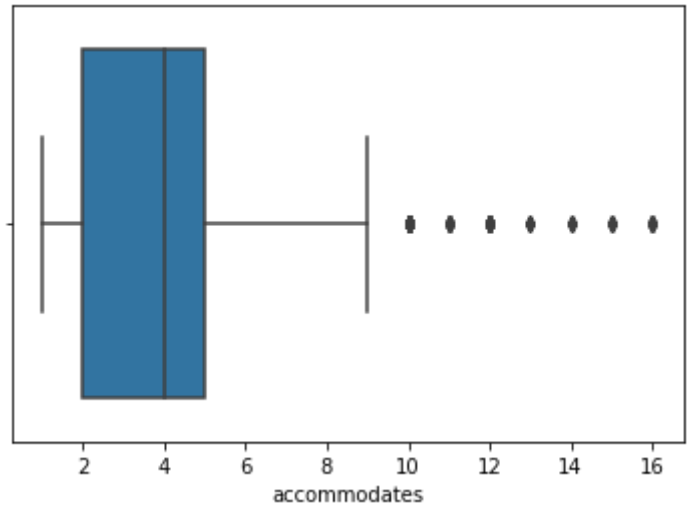


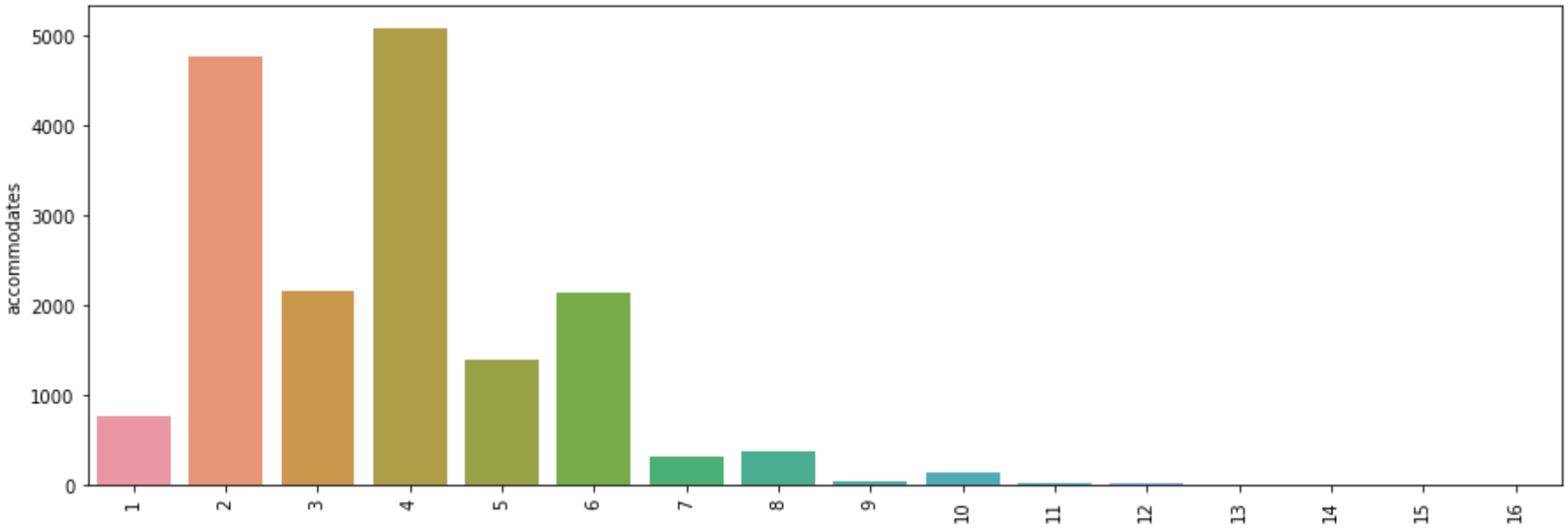


accommodates

In [29]:
```
diagrama_caixa(data['accommodates'])
grafico_barra(data['accommodates'])
data.shape
```

Out[29]: (17232, 17)





In [30]:

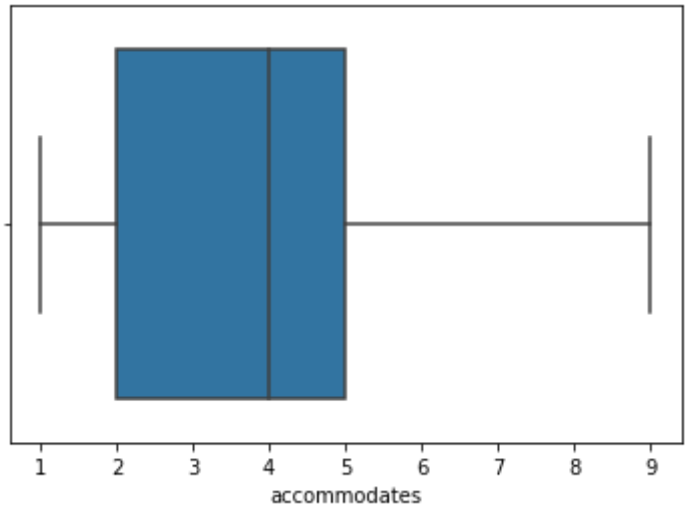
```
# EXCLUINDO OUTLIERS

Q1 = data['accommodates'].quantile(.25)
Q3 = data['accommodates'].quantile(.75)
IQR = Q3 - Q1
LimI = Q1 - (1.5 * IQR)
LimSP = Q3 + (1.5 * IQR)
```

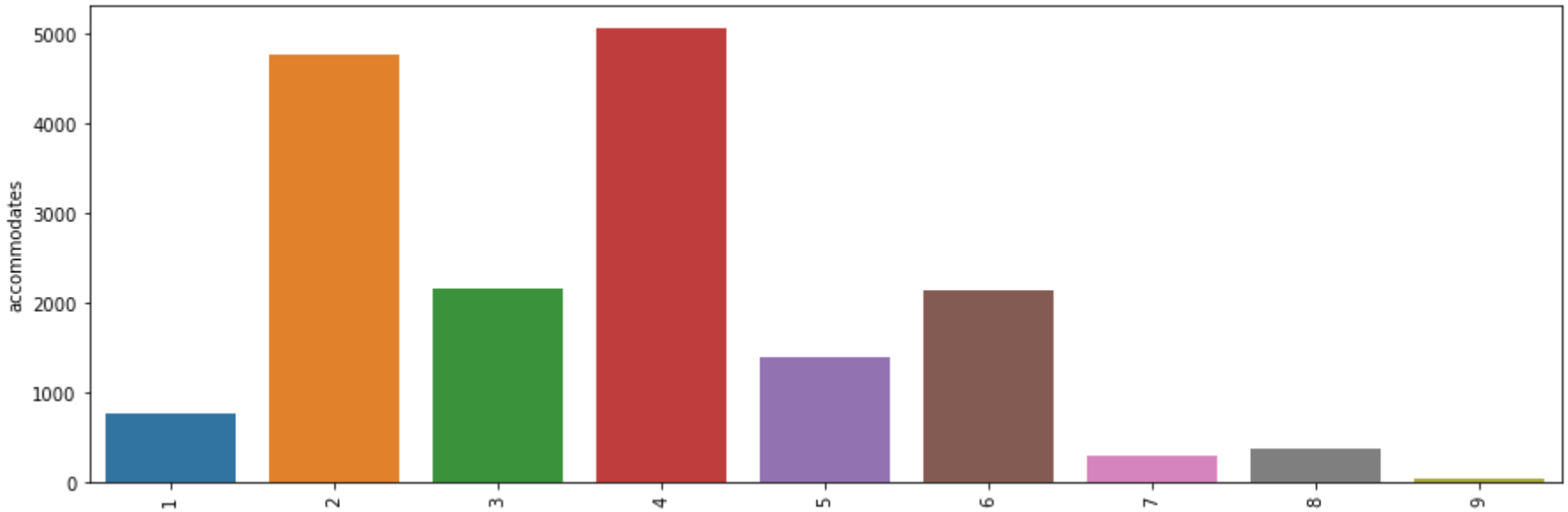
```
data = data.loc[(data['accommodates'] >= LimI) & (data['accommodates'] <= LimSP)]
data.shape
```

Out[30]: (17048, 17)

In [31]:

```
diagrama_caixa(data['accommodates'])
grafico_barra(data['accommodates'])
```

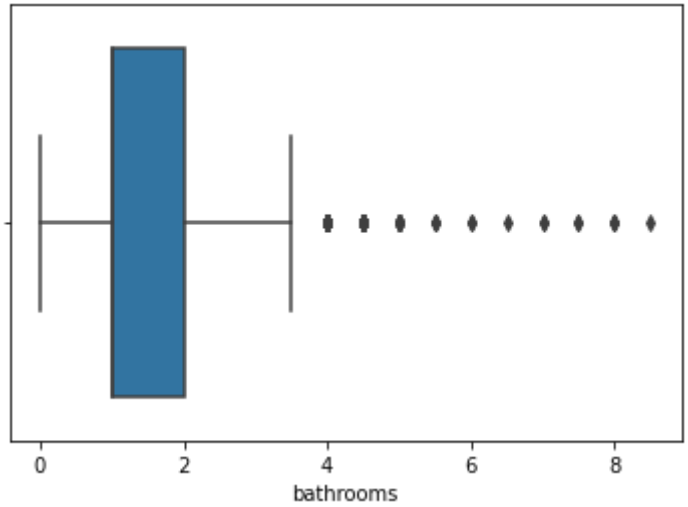


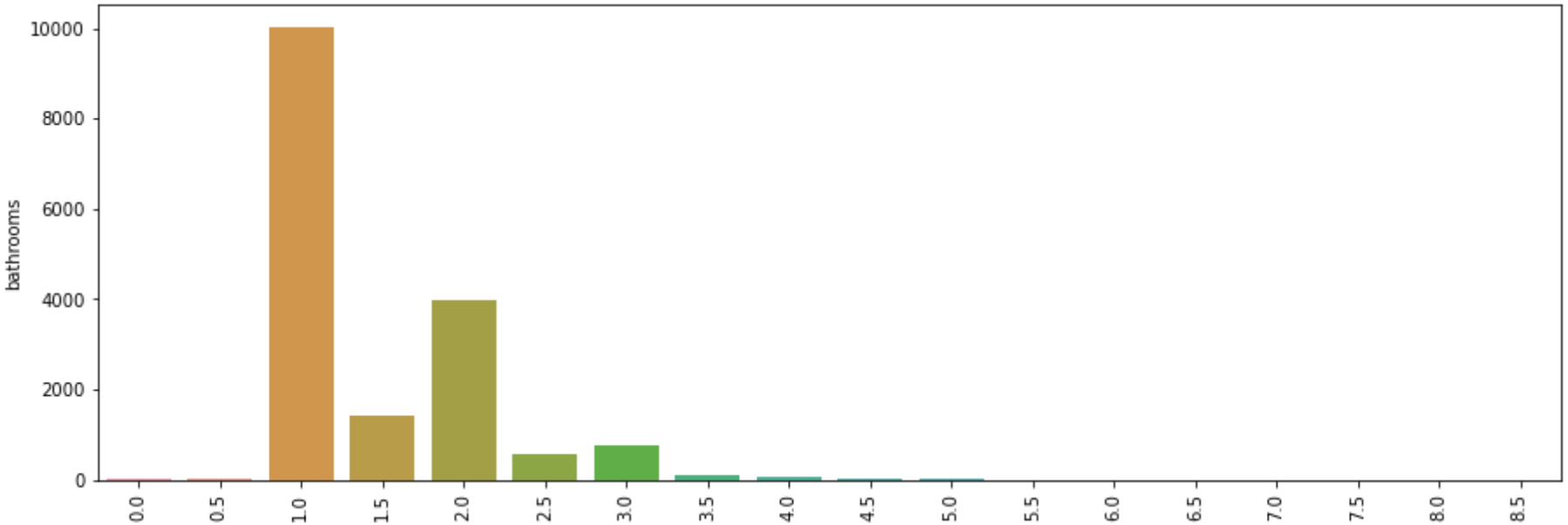


bathrooms

In [32]:
```
diagrama_caixa(data['bathrooms'])
grafico_barra(data['bathrooms'])
data.shape
```

Out[32]: (17048, 17)





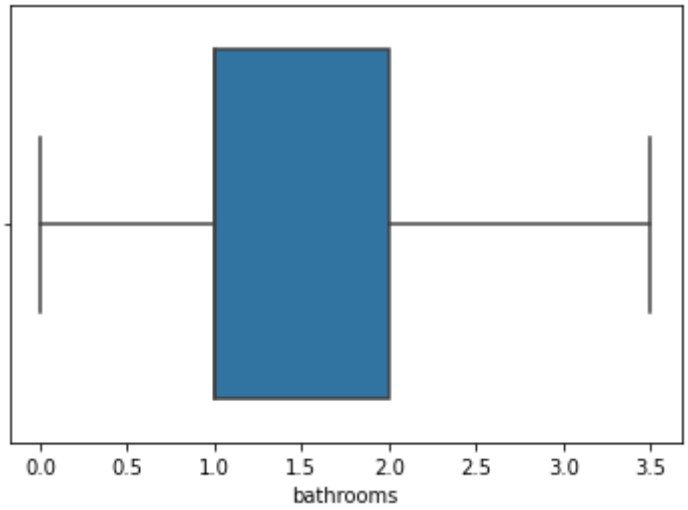
In [33]:

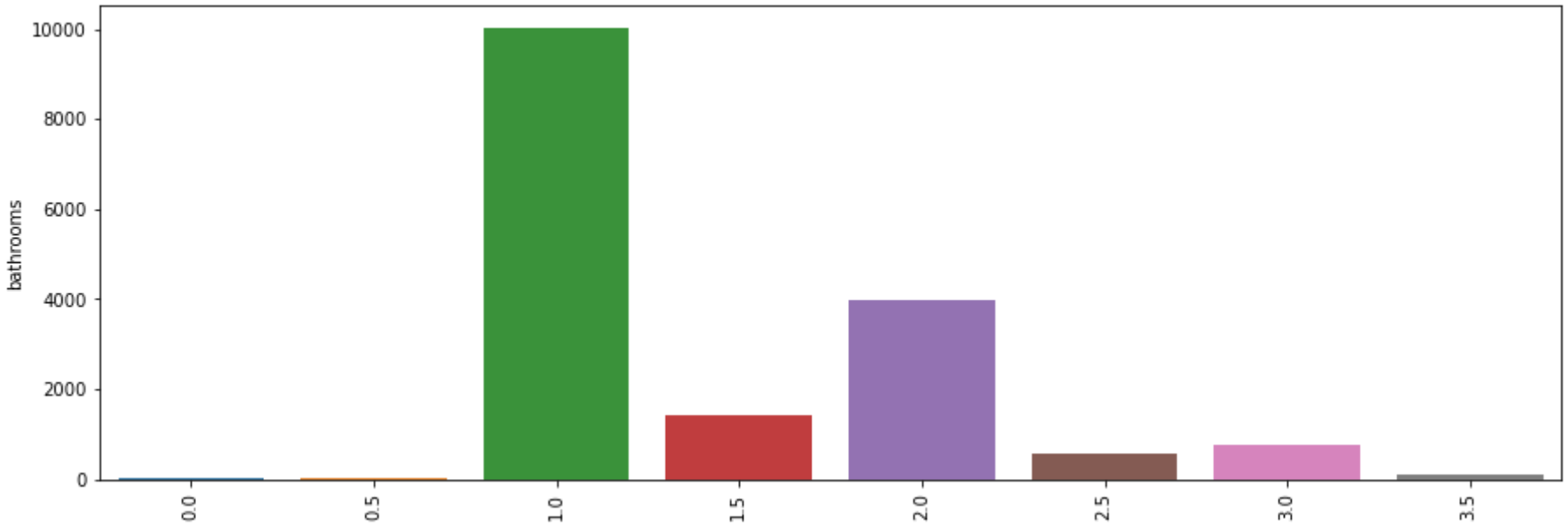
```
# EXCLUINDO OUTLIERS

Q1 = data['bathrooms'].quantile(.25)
Q3 = data['bathrooms'].quantile(.75)
IQR = Q3 - Q1
LimI = Q1 - (1.5 * IQR)
```

```
LimSP = Q3 + (1.5 * IQR)
data = data.loc[(data['bathrooms'] >= LimI) & (data['bathrooms'] <= LimSP)]
data.shape
```

Out[33]: (16928, 17)

In [34]:
```
diagrama_caixa(data['bathrooms'])
grafico_barra(data['bathrooms'])
```

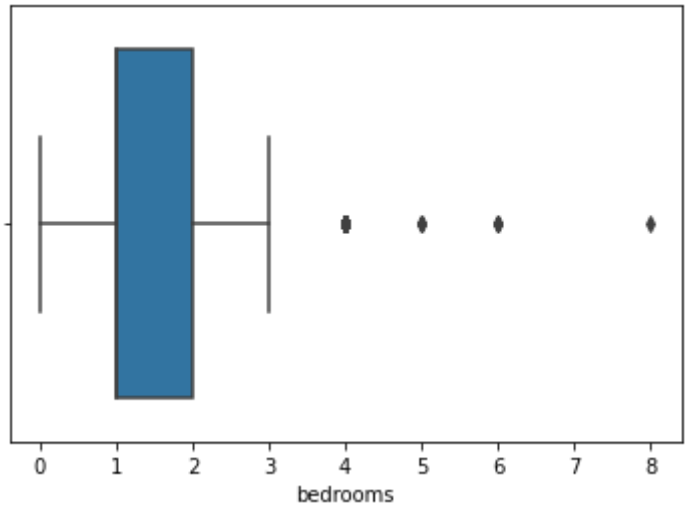


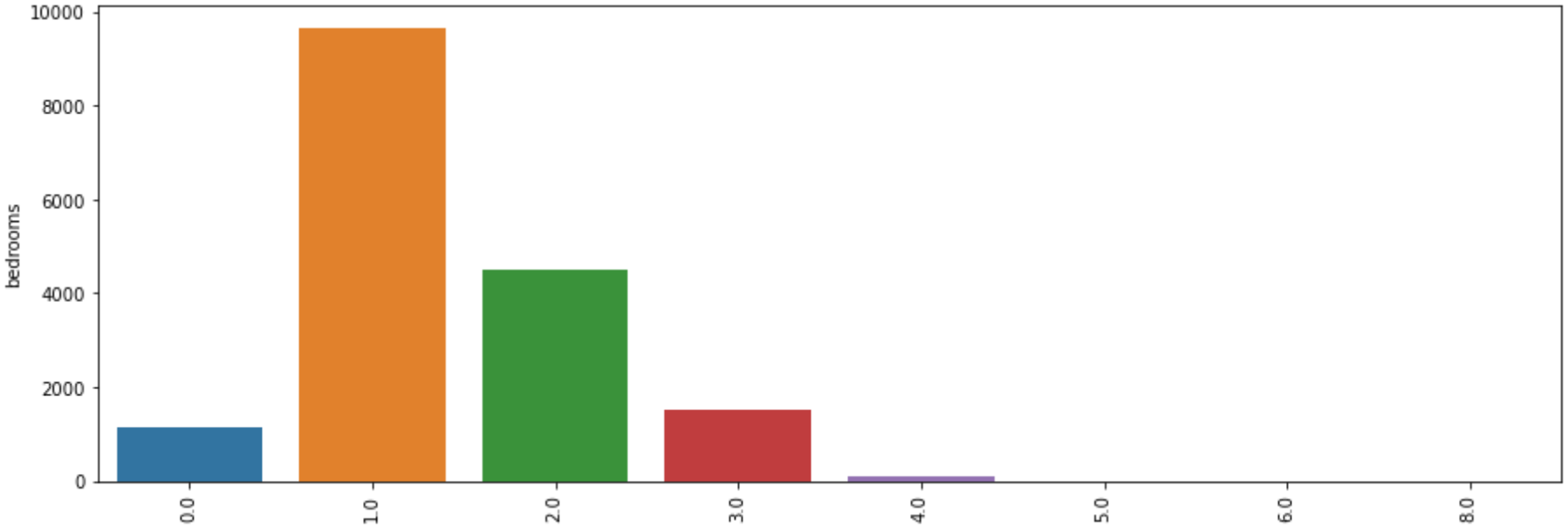


bedrooms

In [35]:
```
diagrama_caixa(data['bedrooms'])
grafico_barra(data['bedrooms'])
data.shape
```

Out[35]: (16928, 17)





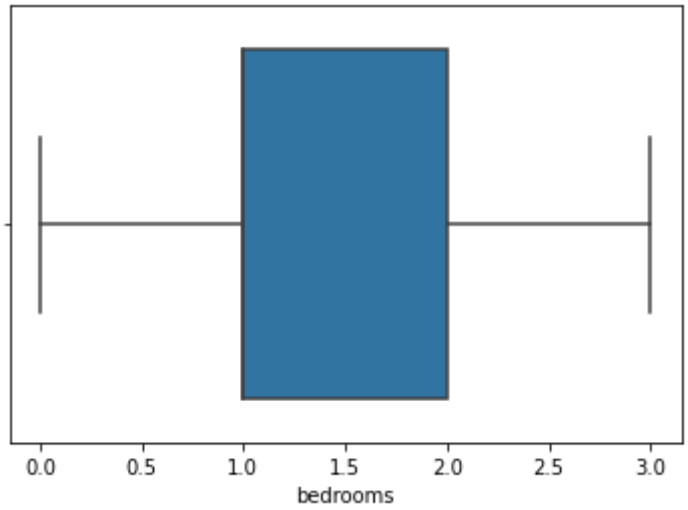
In [36]:

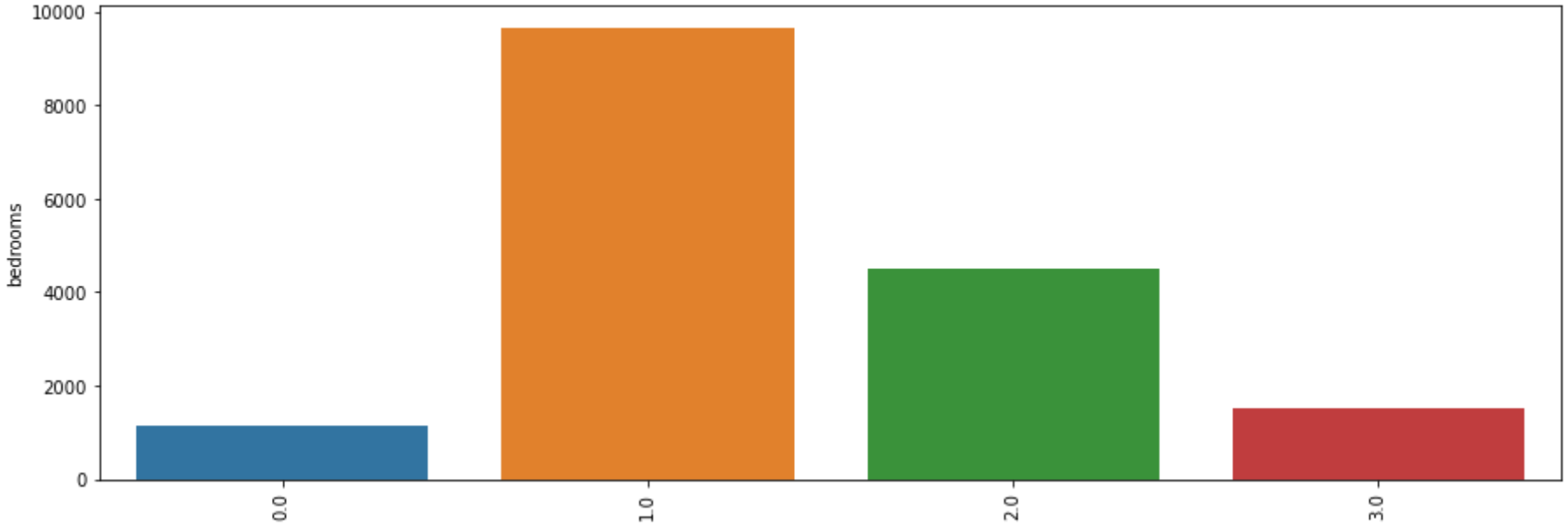
```
# EXCLUINDO OUTLIERS

Q1 = data['bedrooms'].quantile(.25)
Q3 = data['bedrooms'].quantile(.75)
IQR = Q3 - Q1
LimI = Q1 - (1.5 * IQR)
```

```
LimSP = Q3 + (1.5 * IQR)
data = data.loc[(data['bedrooms'] >= LimI) & (data['bedrooms'] <= LimSP)]
data.shape
```

Out[36]: (16822, 17)

In [37]:
```
diagrama_caixa(data['bedrooms'])
grafico_barra(data['bedrooms'])
```





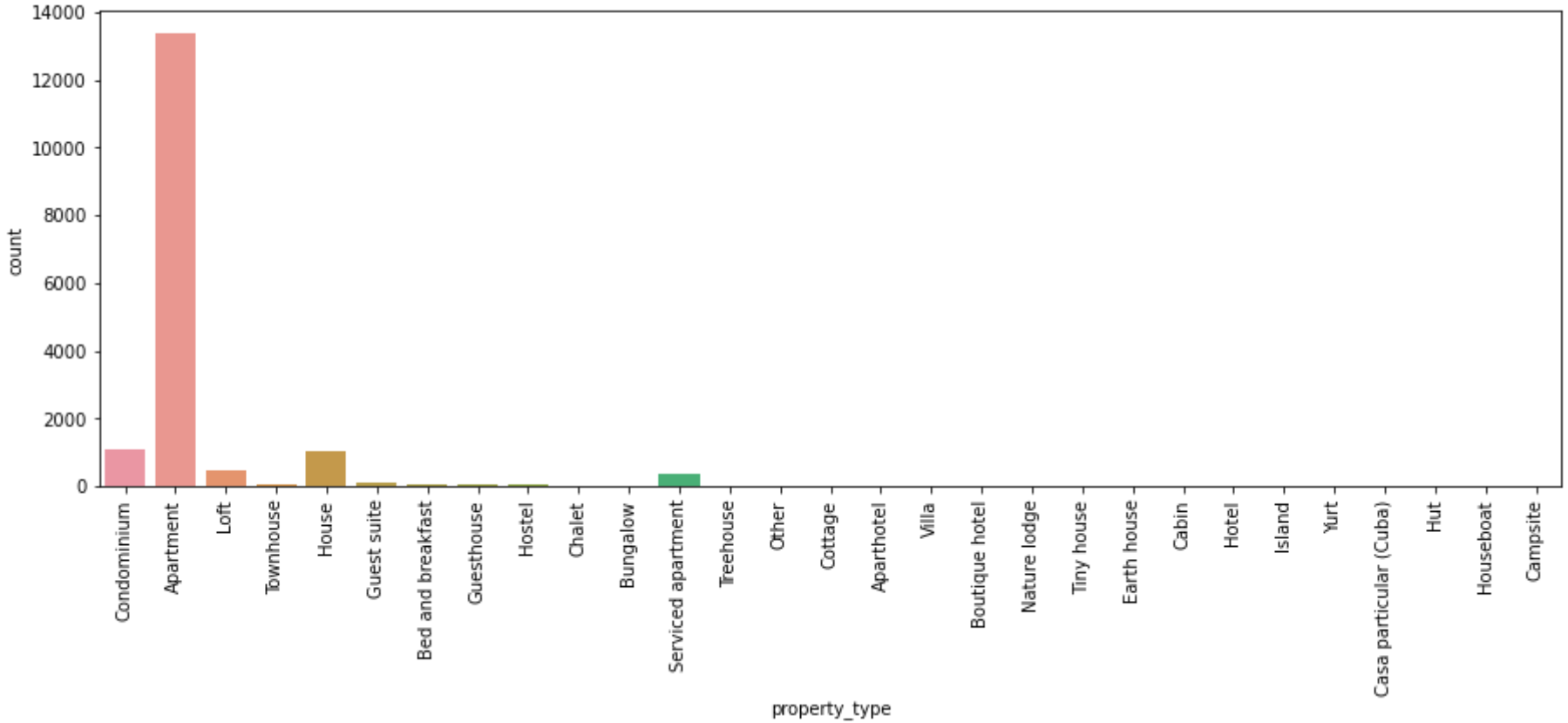
Analisando as colunas com valores de texto

Podemos utilizar critérios para agrupar valores pequenos em um único grupo

property_type

In [38]:
```
plt.figure(figsize=(15, 5))
grafico = sns.countplot('property_type', data=data)
grafico.tick_params(axis='x', rotation=90)
```

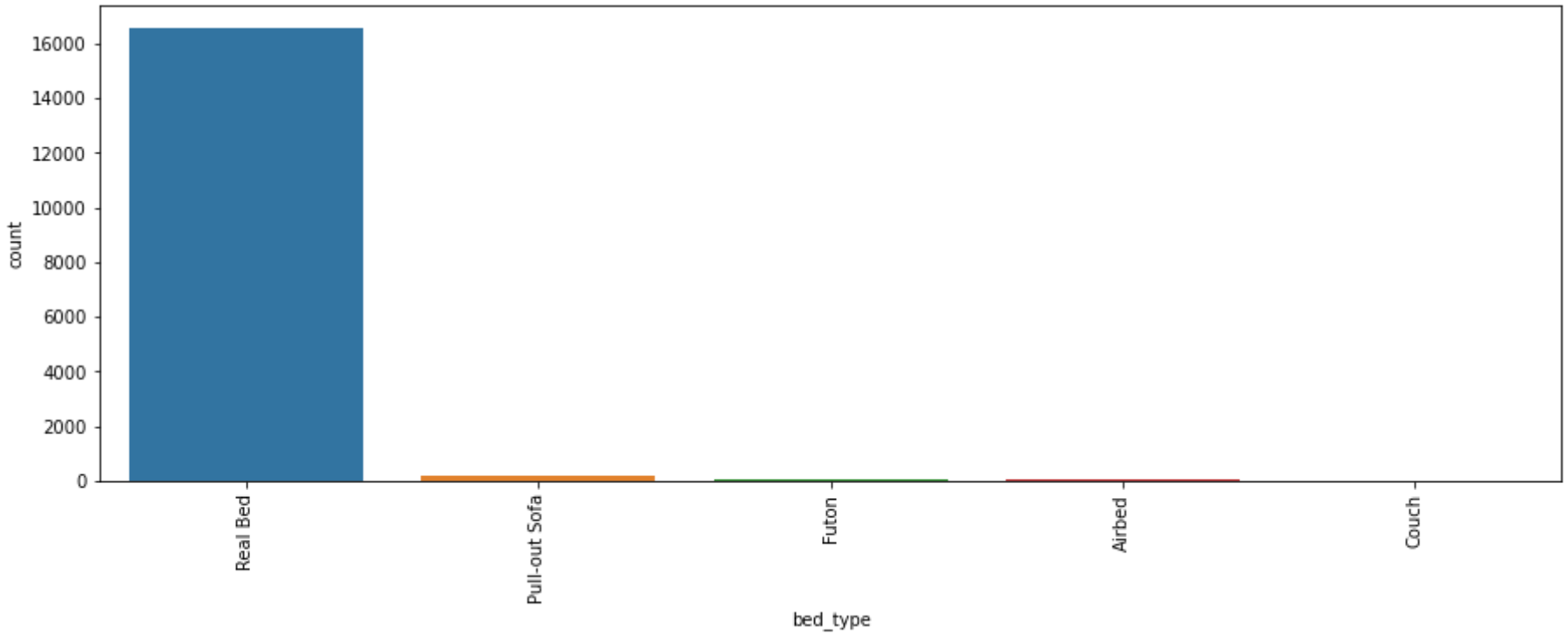
```
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From vers
ion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result i
n an error or misinterpretation.
  warnings.warn(
```



bed_type

In [39]:
```
plt.figure(figsize=(15, 5))
grafico = sns.countplot('bed_type', data=data)
grafico.tick_params(axis='x', rotation=90)
```

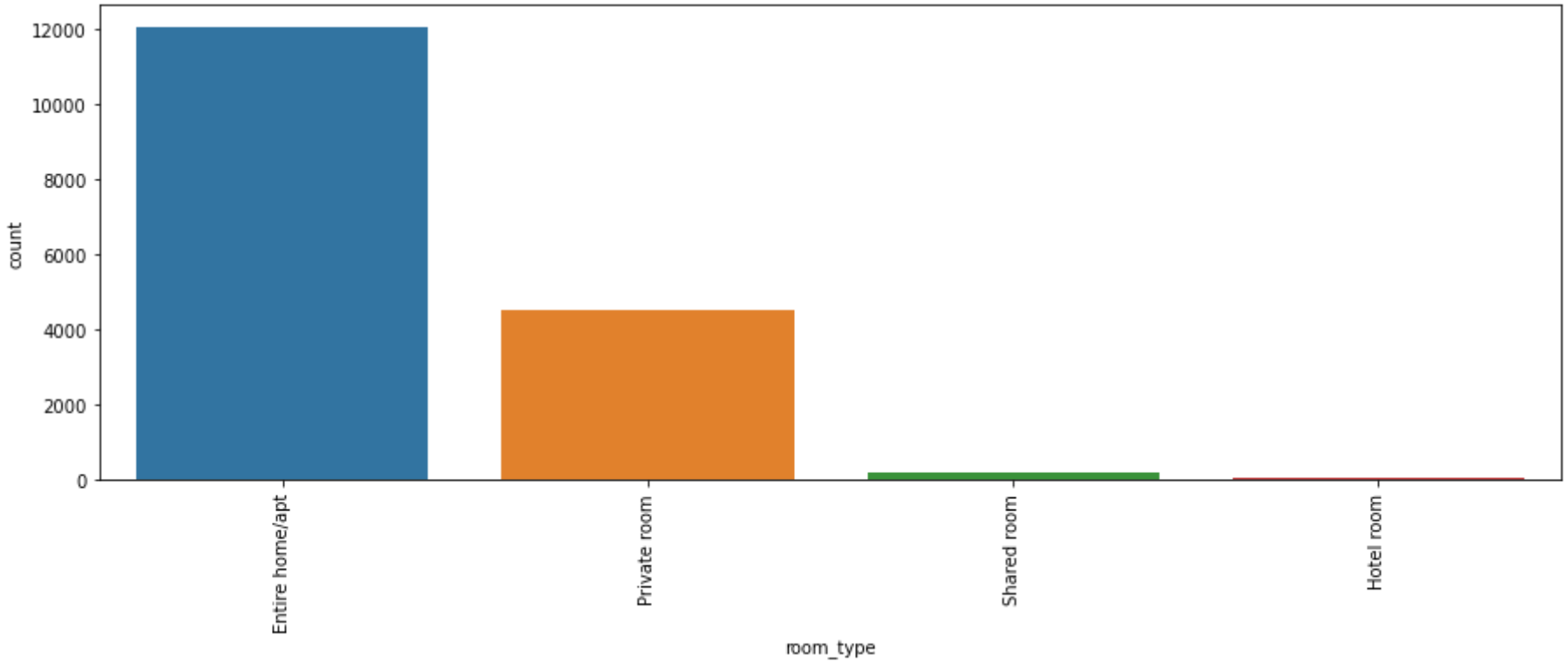
```
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From vers
ion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result i
n an error or misinterpretation.
  warnings.warn(
```



room_type

In [40]:
```
plt.figure(figsize=(15, 5))
grafico = sns.countplot('room_type', data=data)
grafico.tick_params(axis='x', rotation=90)
```

```
C:\Anaconda\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From vers
ion 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result i
n an error or misinterpretation.
  warnings.warn(
```



In [41]:
```
data
```

Out[41]:

	host_listings_count	latitude	longitude	property_type	room_type	accommodates	bathrooms	bedrooms	beds	bed_type	price	cleaning_fee
0	2.0	-22.965920	-43.178960	Condominium	Entire home/apt	5	1.0	2.0	2.0	Real Bed	332.0	378.0
1	0.0	-22.984670	-43.196110	Apartment	Entire home/apt	6	2.0	2.0	4.0	Real Bed	336.0	210.0
2	3.0	-22.977120	-43.190450	Apartment	Entire home/apt	2	1.0	1.0	2.0	Real Bed	159.0	250.0
3	1.0	-22.983020	-43.214270	Apartment	Entire home/apt	3	1.0	1.0	2.0	Real Bed	273.0	84.0
4	1.0	-22.988160	-43.193590	Apartment	Entire home/apt	2	1.5	1.0	1.0	Real Bed	378.0	172.0

	host_listings_count	latitude	longitude	property_type	room_type	accommodates	bathrooms	bedrooms	beds	bed_type	price	cleaning_fee
...	...	...	...	...	...	...	...	...	...	...	...	...
34729	4.0	-22.923567	-43.178353	Apartment	Entire home/apt	2	1.0	0.0	2.0	Real Bed	348.0	100.0
34737	0.0	-22.956178	-43.197239	Apartment	Private room	3	1.0	1.0	1.0	Real Bed	101.0	30.0
34746	2.0	-22.988260	-43.192360	Apartment	Private room	3	3.0	1.0	2.0	Real Bed	470.0	200.0
34749	3.0	-22.984615	-43.190297	Apartment	Private room	2	2.0	1.0	1.0	Real Bed	151.0	50.0
34753	7.0	-22.963840	-43.181151	Apartment	Private room	2	1.0	1.0	1.0	Real Bed	180.0	150.0

16822 rows × 17 columns

Codificando as variáveis

In [42]:
```
colunas_categorias = ['property_type', 'room_type', 'bed_type']
data = pd.get_dummies(data=data, columns=colunas_categorias)
display(data.head())
```

	host_listings_count	latitude	longitude	accommodates	bathrooms	bedrooms	beds	price	cleaning_fee	guests_included	...	property_type_Yurt	roo
0	2.0	-22.96592	-43.17896	5	1.0	2.0	2.0	332.0	378.0	2	...	0	
1	0.0	-22.98467	-43.19611	6	2.0	2.0	4.0	336.0	210.0	6	...	0	
2	3.0	-22.97712	-43.19045	2	1.0	1.0	2.0	159.0	250.0	2	...	0	
3	1.0	-22.98302	-43.21427	3	1.0	1.0	2.0	273.0	84.0	2	...	0	
4	1.0	-22.98816	-43.19359	2	1.5	1.0	1.0	378.0	172.0	2	...	0	

5 rows × 52 columns

In [43]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data[['host_listings_count', 'latitude', 'longitude', 'accommodates', 'bathrooms', 'bedrooms', 'beds', 'cleaning_fee', 'guests_inc
data
```

Out[43]:

	host_listings_count	latitude	longitude	accommodates	bathrooms	bedrooms	beds	price	cleaning_fee	guests_included	...	property_type_Y
0	0.250	0.319118	0.872808	0.500	0.285714	0.666667	0.333333	332.0	0.152727	0.066667	...	
1	0.000	0.259888	0.843465	0.625	0.571429	0.666667	0.666667	336.0	0.084848	0.333333	...	
2	0.375	0.283738	0.853149	0.125	0.285714	0.333333	0.333333	159.0	0.101010	0.066667	...	
3	0.125	0.265100	0.812394	0.250	0.285714	0.333333	0.333333	273.0	0.033939	0.066667	...	
4	0.125	0.248863	0.847777	0.125	0.428571	0.333333	0.166667	378.0	0.069495	0.066667	...	
...	...	...	...	...	...	...	...	...	...	...	...	
34729	0.500	0.452909	0.873846	0.125	0.285714	0.000000	0.333333	348.0	0.040404	0.133333	...	
34737	0.000	0.349892	0.841533	0.250	0.285714	0.333333	0.166667	101.0	0.012121	0.066667	...	
34746	0.250	0.248547	0.849881	0.250	0.857143	0.333333	0.333333	470.0	0.080808	0.066667	...	
34749	0.375	0.260062	0.853411	0.125	0.571429	0.333333	0.166667	151.0	0.020202	0.000000	...	
34753	0.875	0.325690	0.869059	0.125	0.285714	0.333333	0.166667	180.0	0.060606	0.000000	...	

16822 rows × 52 columns

Dividir dados em teste e treino

In [44]:
```
modelo_lr = LinearRegression()
modelo_et = ExtraTreesRegressor()
modelo_rf = RandomForestRegressor()

y = data['price']
```

```
X = data.drop('price', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

Avaliar qual melhor modelo

In [45]:
```
#LinearRegression()

modelo_lr.fit(X_train, y_train)
previsao = modelo_lr.predict(X_test)
r2 = r2_score(y_test, previsao)
RSME = np.sqrt(mean_squared_error(y_test, previsao))
print('r2: {:.2%}'.format(r2))
print('RSME: {:.2f}'.format(RSME))
```

```
r2: 32.11%
RSME: 174.13
```

In [46]:
```
#ExtraTreesRegressor()

modelo_et.fit(X_train, y_train)
previsao = modelo_et.predict(X_test)
r2 = r2_score(y_test, previsao)
RSME = np.sqrt(mean_squared_error(y_test, previsao))
print('r2: {:.2%}'.format(r2))
print('RSME: {:.2f}'.format(RSME))
```

```
r2: 42.73%
RSME: 159.93
```

In [47]:
```
#RandomForestRegressor()

modelo_rf.fit(X_train, y_train)
previsao = modelo_rf.predict(X_test)
r2 = r2_score(y_test, previsao)
RSME = np.sqrt(mean_squared_error(y_test, previsao))
print('r2: {:.2%}'.format(r2))
print('RSME: {:.2f}'.format(RSME))
```

```
r2: 45.72%
RSME: 155.70
```

Resultado

Analisando o resultado, iremos adotar o RandomForestRegressor() como melhor modelo, \ Levando em consideração seu Coeficiente de Correlação maior (r^2) \ E Erro quadrático médio (RMSE) menor