Importing necessary modules

In [1]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from tabulate import tabulate
```

In [2]:

```python
empdf=pd.read_csv(r"K:\Desktop\NIIT\tables\DS1_C5_S3_Employee_Data_Concept.csv")
empdf
```

Out[2]:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
0	São Paulo	70	2	1	1	7	acept	furnished	2065	3300	211	42	5618
1	São Paulo	320	4	4	0	20	acept	not furnished	1200	4960	1750	63	7973
2	Porto Alegre	80	1	1	1	6	acept	not furnished	1000	2800	0	41	3841
3	Porto Alegre	51	2	1	0	2	acept	not furnished	270	1112	22	17	1421
4	São Paulo	25	1	1	0	1	not acept	not furnished	0	800	25	11	836
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10687	Porto Alegre	63	2	1	1	5	not acept	furnished	402	1478	24	22	1926
10688	São Paulo	285	4	4	4	17	acept	not furnished	3100	15000	973	191	19260
10689	Rio de Janeiro	70	3	3	0	8	not acept	furnished	980	6000	332	78	7390
10690	Rio de Janeiro	120	2	2	2	8	acept	furnished	1585	12000	279	155	14020
10691	São Paulo	80	2	1	0	0	acept	not furnished	0	1400	165	22	1587

10692 rows × 13 columns

Level 0 : Data Exploration

1.Visually inspect the first few and last few rows of the data

In [3]:

```python
empdf.head()
```

Out[3]:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
0	São Paulo	70	2	1	1	7	acept	furnished	2065	3300	211	42	5618
1	São Paulo	320	4	4	0	20	acept	not furnished	1200	4960	1750	63	7973
2	Porto Alegre	80	1	1	1	6	acept	not furnished	1000	2800	0	41	3841
3	Porto Alegre	51	2	1	0	2	acept	not furnished	270	1112	22	17	1421
4	São Paulo	25	1	1	0	1	not acept	not furnished	0	800	25	11	836

In [4]:

```python
empdf.tail()
```

Out[4]:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	fire insurance (R$)	total (R$)
10687	Porto Alegre	63	2	1	1	5	not acept	furnished	402	1478	24	22	1926
10688	São Paulo	285	4	4	4	17	acept	not furnished	3100	15000	973	191	19260
10689	Rio de Janeiro	70	3	3	0	8	not acept	furnished	980	6000	332	78	7390
10690	Rio de Janeiro	120	2	2	2	8	acept	furnished	1585	12000	279	155	14020
10691	São Paulo	80	2	1	0	0	acept	not furnished	0	1400	165	22	1587

2.Check the shape of the data frame

In [5]:

```python
print("Number of rows and columns = ",empdf.shape)
```

```
Number of rows and columns =  (10692, 13)
```

3.Check the count of null values in each column

In [6]:

```python
print(empdf.isnull().sum())
print()
print("No missing values ")
```

```
city                   0
area                   0
rooms                  0
bathroom               0
parking spaces         0
floor                  0
animal                 0
furniture              0
hoa (R$)               0
rent amount (R$)       0
property tax (R$)      0
fire insurance (R$)    0
total (R$)             0
dtype: int64

No missing values
```

4.Inspect all the column names and cross check with the data dictionary

In [7]:

```python
empdf.columns
```

Out[7]:

```
Index(['city', 'area', 'rooms', 'bathroom', 'parking spaces', 'floor',
       'animal', 'furniture', 'hoa (R$)', 'rent amount (R$)',
       'property tax (R$)', 'fire insurance (R$)', 'total (R$)'],
      dtype='object')
```

5.Check the information of the data frame using the info() function

In [8]:

```
empdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10692 entries, 0 to 10691
Data columns (total 13 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   city                 10692 non-null  object
 1   area                 10692 non-null  int64 
 2   rooms                10692 non-null  int64 
 3   bathroom             10692 non-null  int64 
 4   parking spaces       10692 non-null  int64 
 5   floor                10692 non-null  int64 
 6   animal               10692 non-null  object
 7   furniture            10692 non-null  object
 8   hoa (R$)             10692 non-null  int64 
 9   rent amount (R$)     10692 non-null  int64 
 10  property tax (R$)    10692 non-null  int64 
 11  fire insurance (R$)  10692 non-null  int64 
 12  total (R$)           10692 non-null  int64 
dtypes: int64(10), object(3)
memory usage: 1.1+ MB
```

LEVEL 1 Analysis

Identify if the type data in each column is categorical or numerical?

1. Separate out the categorical columns from the numerical types

These are the kind of analyses that can be performed on categorical data

1. Check if it is Nominal or Ordinal
2. Check how many categories are present
3. Check the Mode
4. Check for Missing values
5. Think about how the missing values could be treated
6. Think about the kind of graph/chart that can be plotted using this data

Note: We are analyzing only one column at a time (Univariate Analysis).

In [9]:

```
def seperator(df):
    categorical=[]
    numerical=[]
    for col in df.columns:
        if(df[col].nunique()<100):
            categorical.append(col)
        else:
            numerical.append(col)
    return categorical,numerical

categorical,numerical=seperator(empdf)
print(tabulate({"Categorical":categorical,"continuous": numerical},headers = ["categorical", "numerical"]))
def bar_percentage(ax, count: "number of rows in data "):
    for bar in ax.patches:
        percentage = f"{round((bar.get_height() / count) *100, 2)}%"

        x = bar.get_x() + bar.get_width() /2
        y = bar.get_height()
        ax.annotate(percentage, (x, y), va = "bottom", ha = "center")

def cat_level1(df,col):
        fig,ax=plt.subplots(1,2,figsize=(18,6))
        print("Number of Unique values present = ",df[col].nunique())
        print("NA values = ",df[col].isnull().sum())
        print("Mode = ",df[col].mode()[0])
        df[col].fillna(df[col].mode()[0],inplace=True)
        sns.countplot(x=df[col],ax=ax[0])
        ax[0]=bar_percentage(ax[0], len(df))
        percentage=df[col].value_counts()
        labels=df[col].value_counts().index
        ax[1].pie(percentage,labels = list(labels), autopct= "%0.2f%%")
        ax[1].set_title(col+" compostion")
        plt.show()

def num_level1(df,col):
    print(f"The mean of the {col} is {df[col].mean()}")
    print(f"The median of the {col} is {df[col].median()}")
    print(f"The mode of the {col} is {df[col].mode()[0]}")
    print(f"The standard deviation of the {col} is {df[col].std()}")
    print(f"Number of missing values in the {col} is {df[col].isnull().sum()}")
    fig, ax = plt.subplots(1, 2, figsize= (10,5))
    sns.histplot(x = df[col], ax =ax[0], color = "blue")
    sns.boxplot(x = df[col], ax = ax[1], color = "purple",showmeans=True)
    plt.show()

def outlier_treatment(dataframe,columns):
    for item in columns:
        percentile25 = dataframe[item].quantile(0.25)
        percentile75 = dataframe[item].quantile(0.75)
        iqr=percentile75-percentile25
        upper_limit = percentile75 + 1.5 * iqr
        lower_limit = percentile25 - 1.5 * iqr
        dataframe[item] = np.where(dataframe[item] > upper_limit,upper_limit,
        np.where(dataframe[item] < lower_limit,lower_limit,dataframe[item]))
    return dataframe


```

```
categorical     numerical
--------------  -------------------
city            area
rooms           hoa (R$)
bathroom        rent amount (R$)
parking spaces  property tax (R$)
floor           fire insurance (R$)
animal          total (R$)
furniture
```

Plotting level 1 Analysis on Categorical

In [10]:

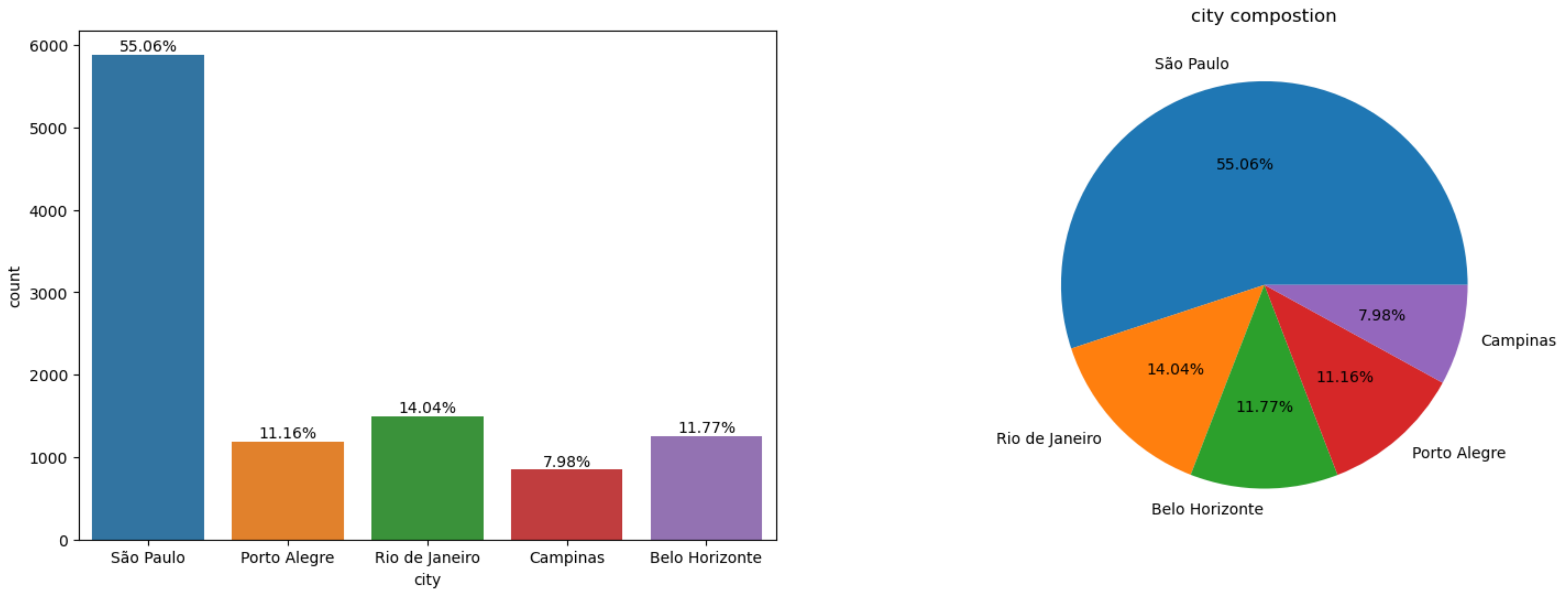
```
categorical
```

Out[10]:

```
['city', 'rooms', 'bathroom', 'parking spaces', 'floor', 'animal', 'furniture']
```

```
In [11]: cat_level1(empdf,"city")
```

```
Number of Unique values present =  5
NA values =  0
Mode =  São Paulo
```

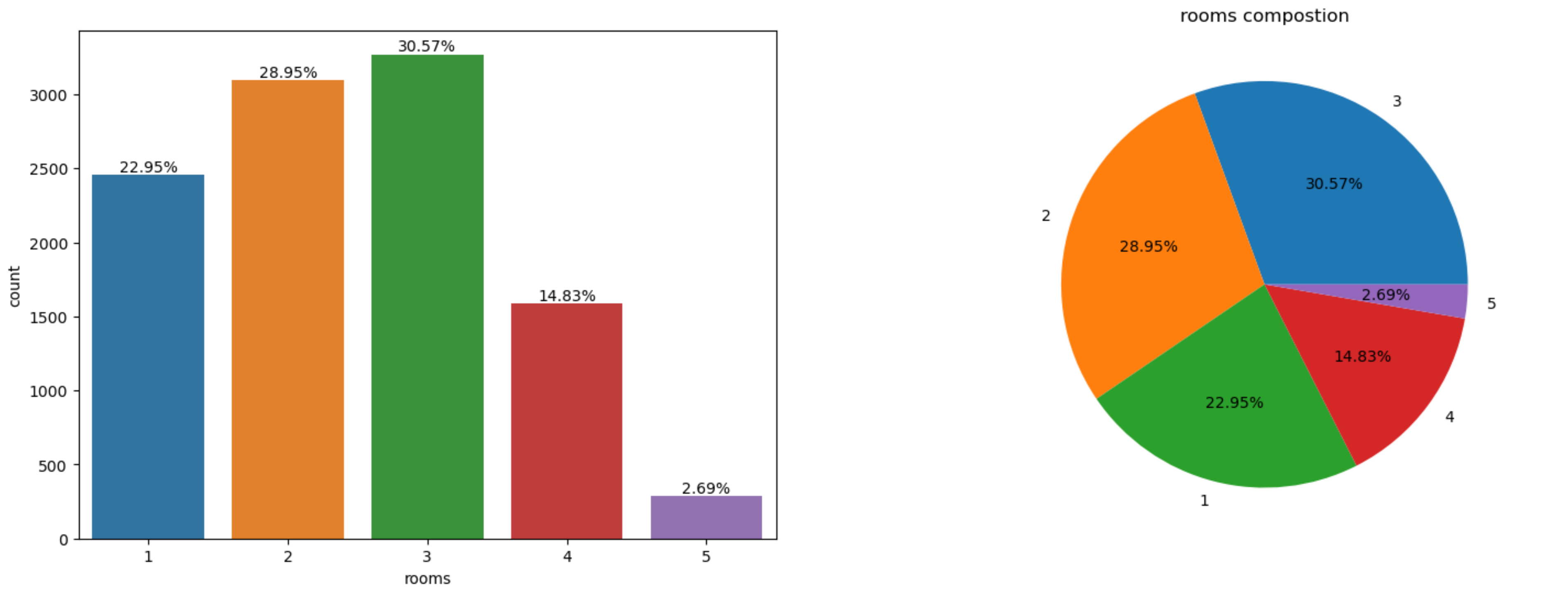


Interpretation :

From above graphs we can see that majority of employees are from Sao Paulo

```
In [12]: mean = int(empdf.rooms.mean())
         x = empdf[empdf["rooms"] > 5].index
         for index in x:
             empdf.loc[index, "rooms"] = mean
         cat_level1(empdf,"rooms")
```

```
Number of Unique values present =  5
NA values =  0
Mode =  3
```



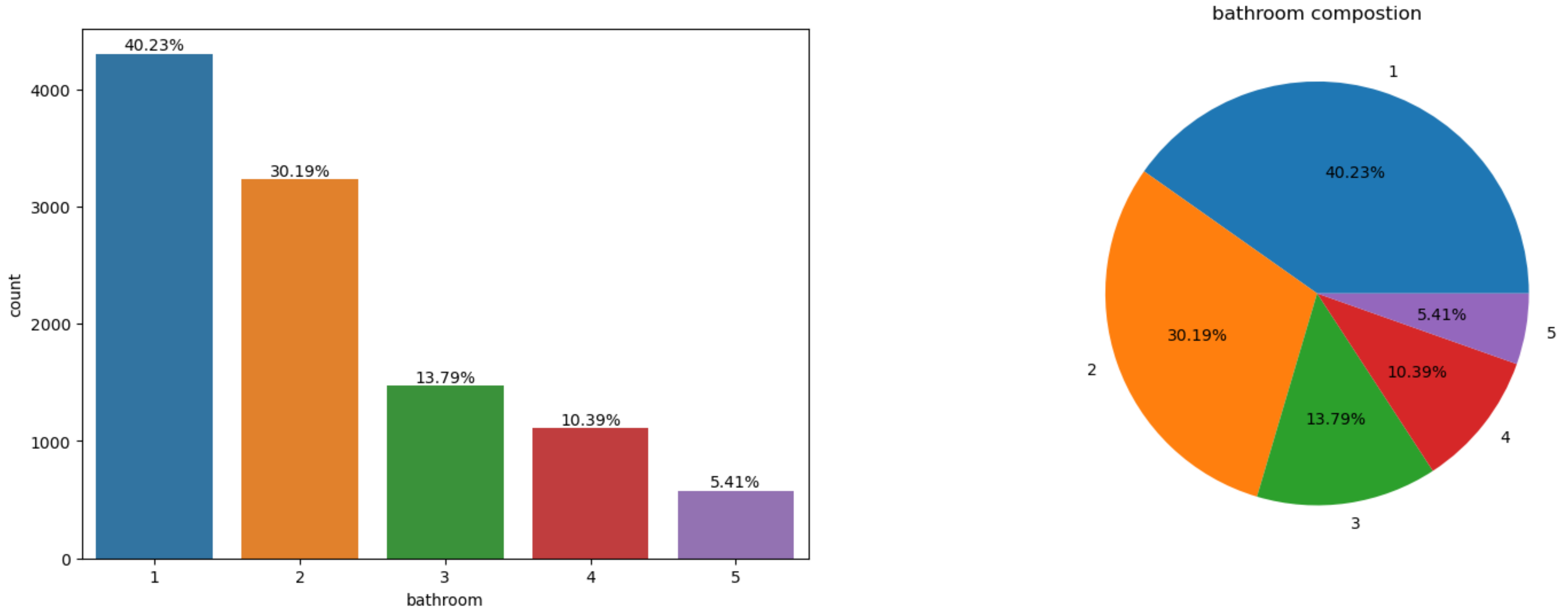
Interpretation :

Majority of the homes are 2 bedroom and 3 bedroom contributing to more than 58% of all composition of homes

In [13]:

```
mean = int(empdf.bathroom.mean())
x = empdf[empdf["bathroom"] > 5].index
for index in x:
    empdf.loc[index, "bathroom"] = mean
cat_level1(empdf,"bathroom")
```

```
Number of Unique values present =  5
NA values =  0
Mode =  1
```



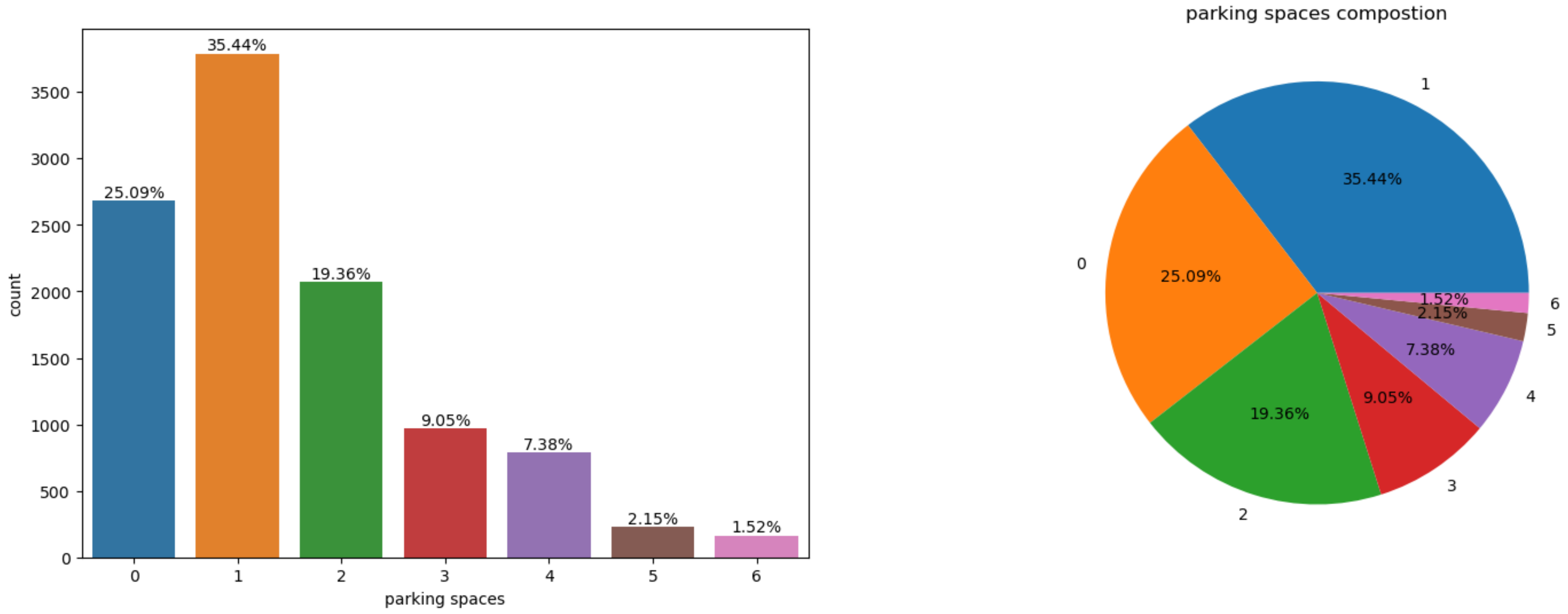
Interpretation :

Majority of homes have 1 and 2 bathrooms contributing 70% of the homes

In [14]:

```
mean = int(empdf["parking spaces"].mean())
x = empdf[empdf["parking spaces"] > 6].index
for index in x:
    empdf.loc[index, "parking spaces"] = mean
cat_level1(empdf,"parking spaces")
```

```
Number of Unique values present =  7
NA values =  0
Mode =  1
```

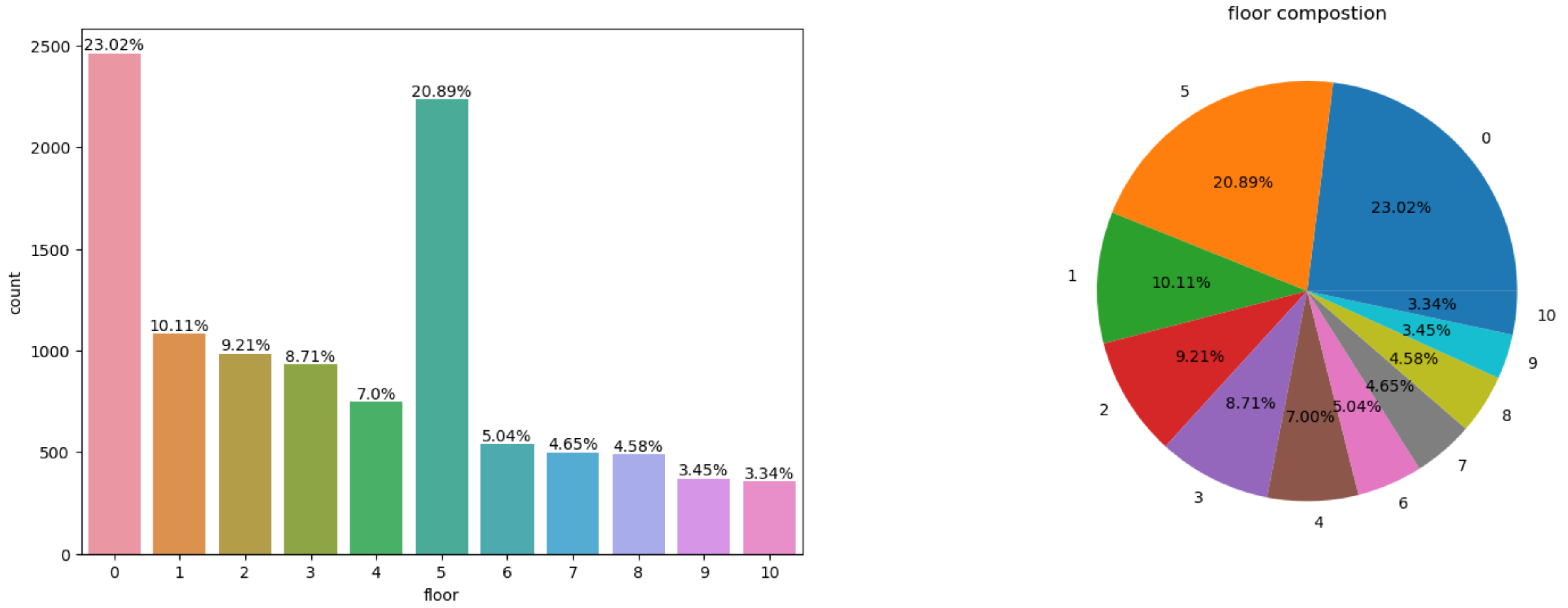


Interpretation :

Most of homes have only 1 parking space contributing to 35% and next to it nearly 25% of homes having no parking space at all

In [15]:
```
mean = int(empdf["floor"].mean())
x = empdf[empdf["floor"] > 10].index
for index in x:
    empdf.loc[index, "floor"] = mean
cat_level1(empdf,"floor")
```

```
Number of Unique values present =  11
NA values =  0
Mode =  0
```

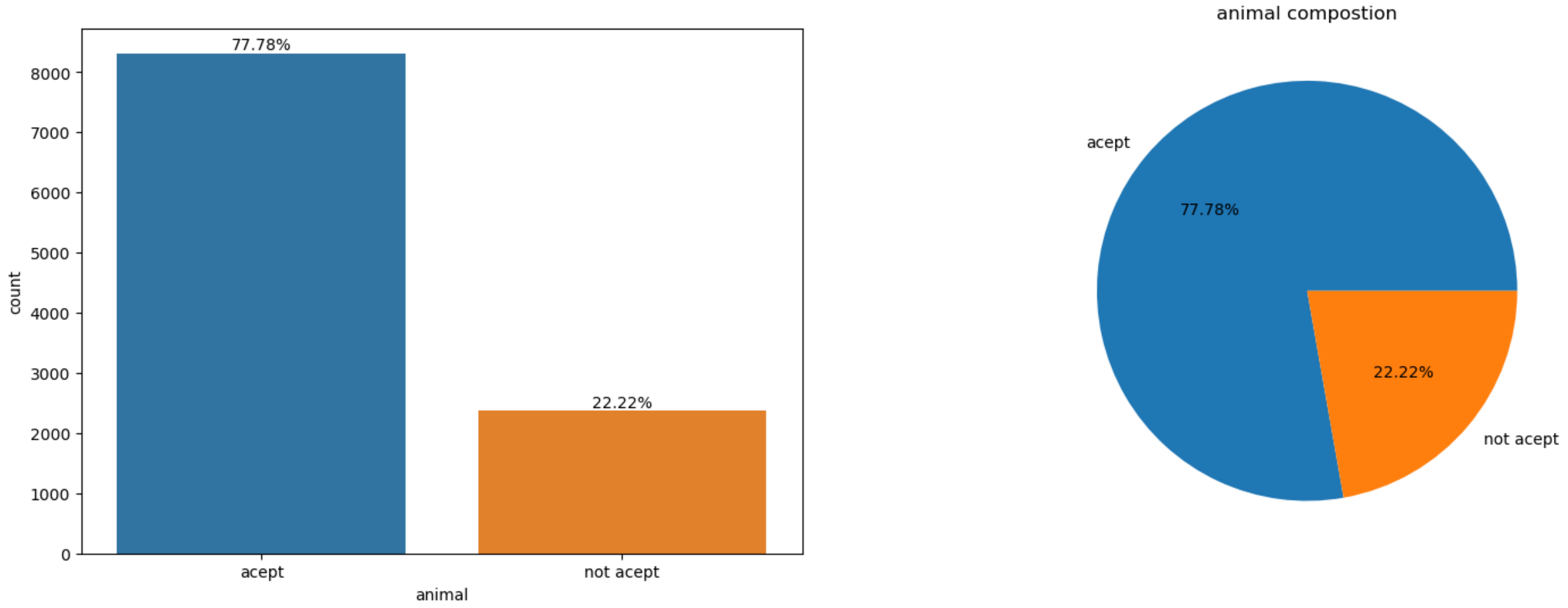


Interpretation:

In many homes there are no extra floors they all are villa .... most homes have 5 floors

In [16]:
```
cat_level1(empdf,"animal")
```

```
Number of Unique values present =  2
NA values =  0
Mode =  acept
```

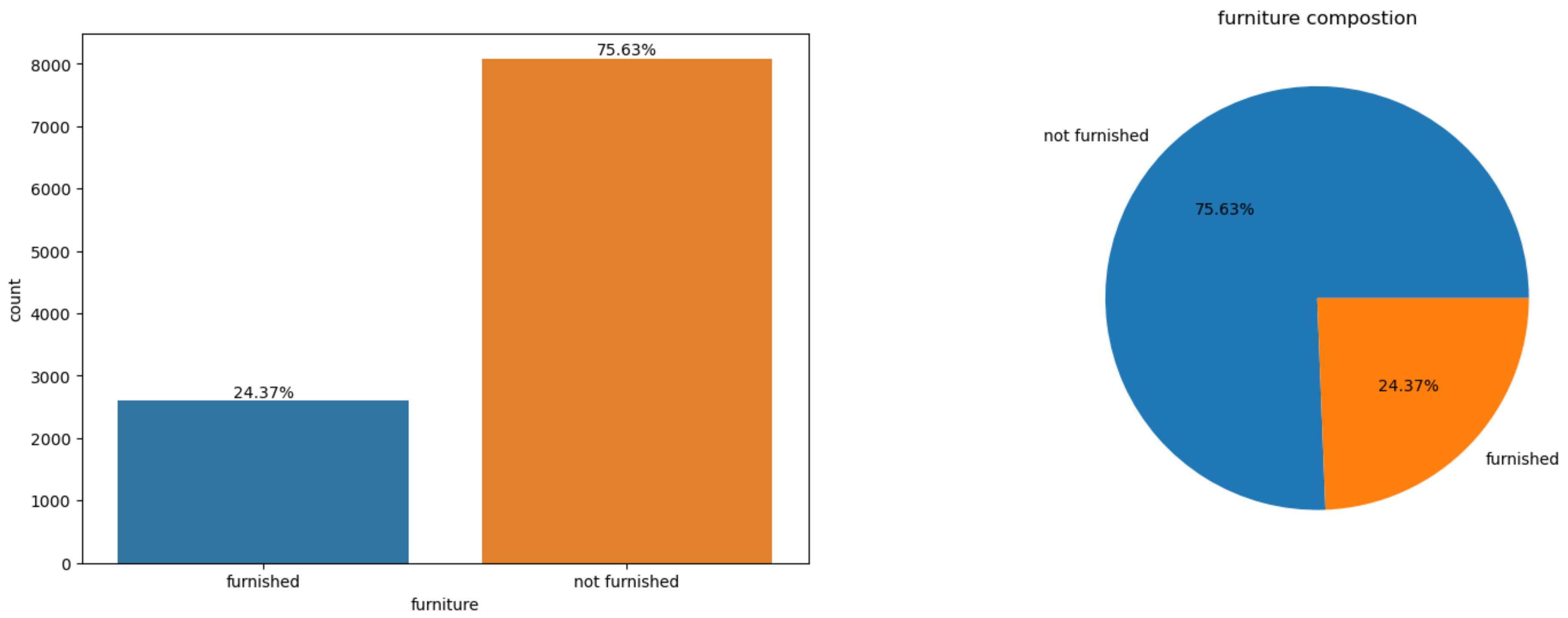


Interpretation

In almost all homes nearly 77% accept pets

```
In [17]: cat_level1(empdf,"furniture")
```

```
Number of Unique values present =  2
NA values =  0
Mode =  not furnished
```



Interpretation :

Nearly 75% homes are unfurnished and 25% are furnished

Level 1 Analysis for numerical data

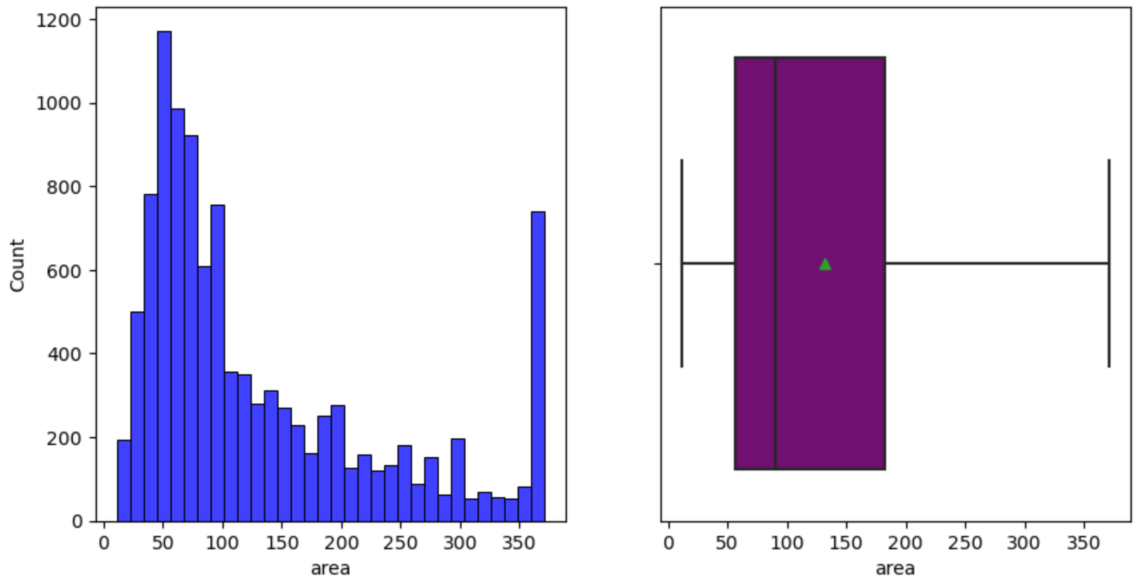
Outlier treatment for all the data in continous

```
In [18]: empdf=outlier_treatment(empdf,numerical)
```

The data has been cleaned of all possible outliers post outlier treatment which can be observed in below boxplots

```
In [19]: num_level1(empdf,numerical[0])
```

```
The mean of the area is 132.0876356154134
The median of the area is 90.0
The mode of the area is 371.0
The standard deviation of the area is 101.33092381207521
Number of missing values in the area is 0
```

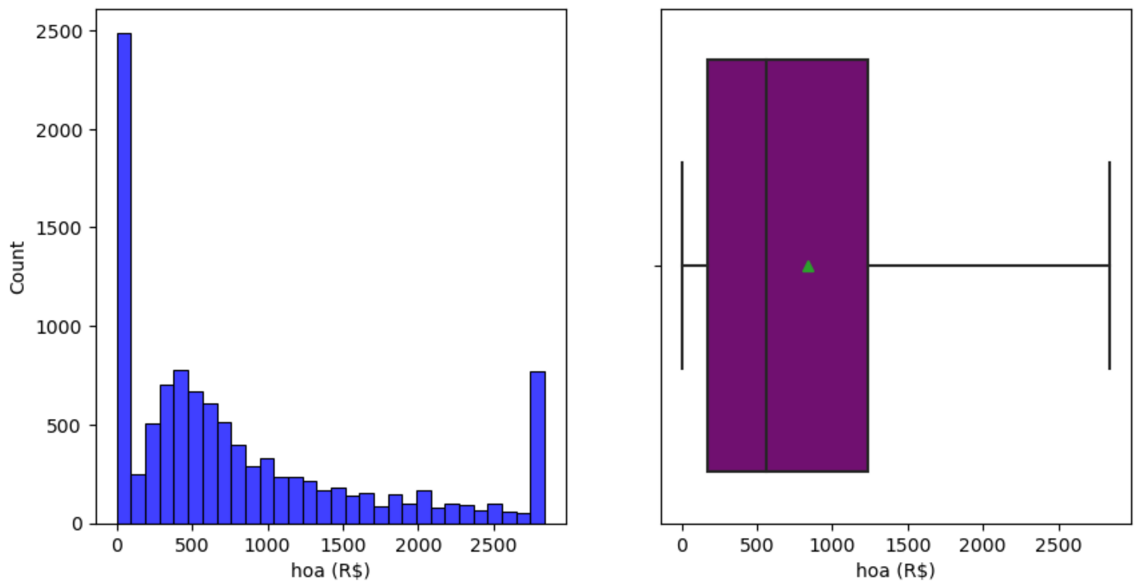


Interpretation:

We can see the boxplot looks clean of outliers as well as majority of data lie in 0 to 150sqft

In [20]:
```
num_level1(empdf,numerical[1])
```

```
The mean of the hoa (R$) is 836.9882856341189
The median of the hoa (R$) is 560.0
The mode of the hoa (R$) is 0.0
The standard deviation of the hoa (R$) is 856.598027516404
Number of missing values in the hoa (R$) is 0
```

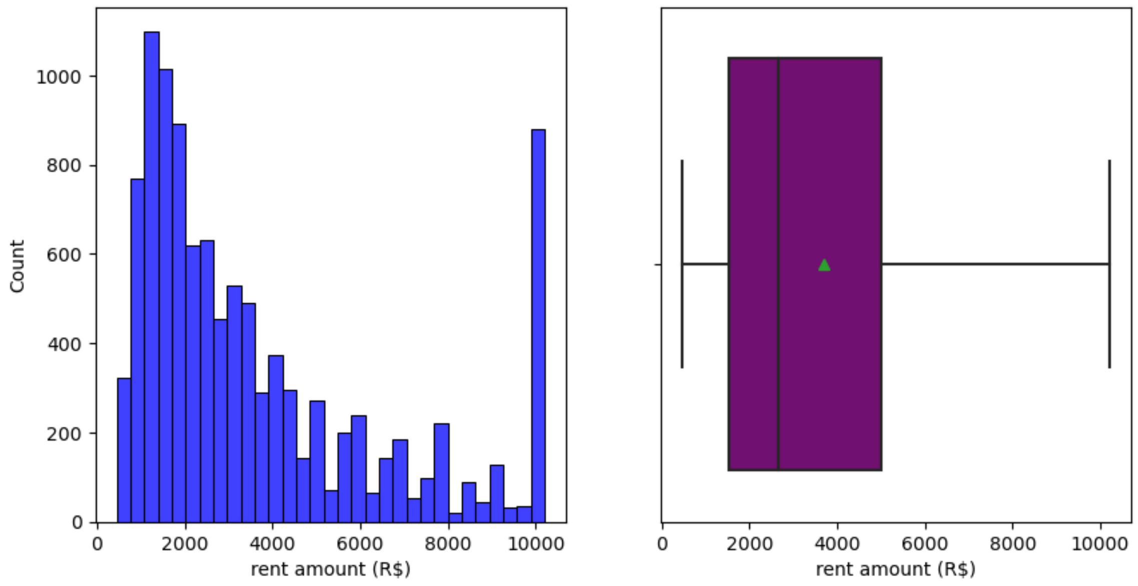


Interpretation :

From the above charts its clear that all the Hoa rates lie in 100 to 1250 it has normal distribution

In [21]:
```
num_level1(empdf,numerical[2])
```

```
The mean of the rent amount (R$) is 3688.2547699214365
The median of the rent amount (R$) is 2661.0
The mode of the rent amount (R$) is 10205.0
The standard deviation of the rent amount (R$) is 2821.8628993304974
Number of missing values in the rent amount (R$) is 0
```

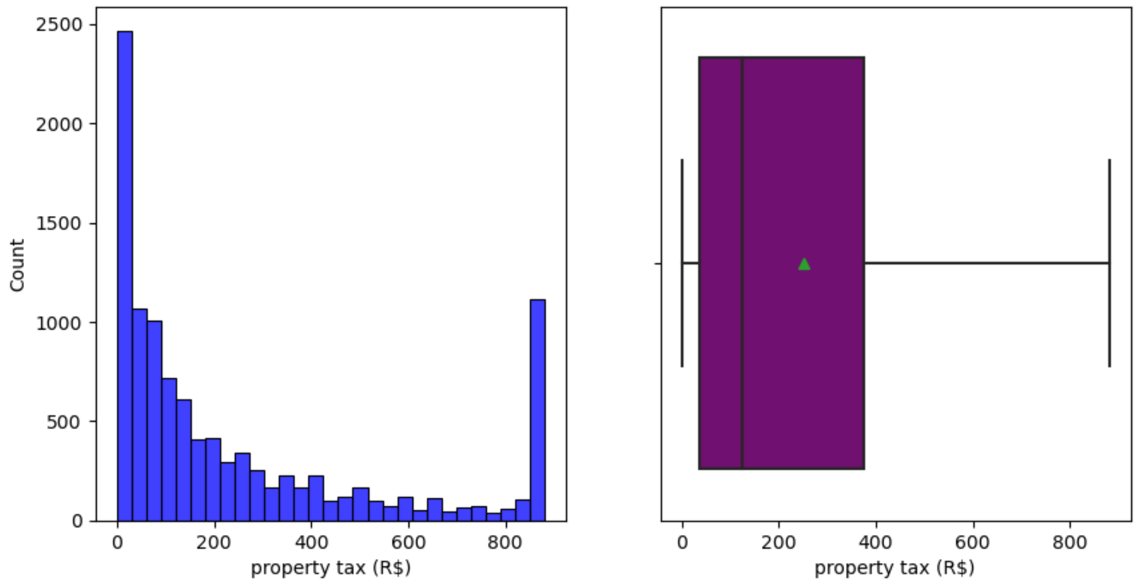


Interpretation:

Here its visible that the highest we can rent a home for is 10k dollars and most homes fall in range of 1200 to 3000 dollars

In [22]:
```
num_level1(empdf,numerical[3])
```

```
The mean of the property tax (R$) is 252.17513093901982
The median of the property tax (R$) is 125.0
The mode of the property tax (R$) is 0.0
The standard deviation of the property tax (R$) is 287.468106240082
Number of missing values in the property tax (R$) is 0
```



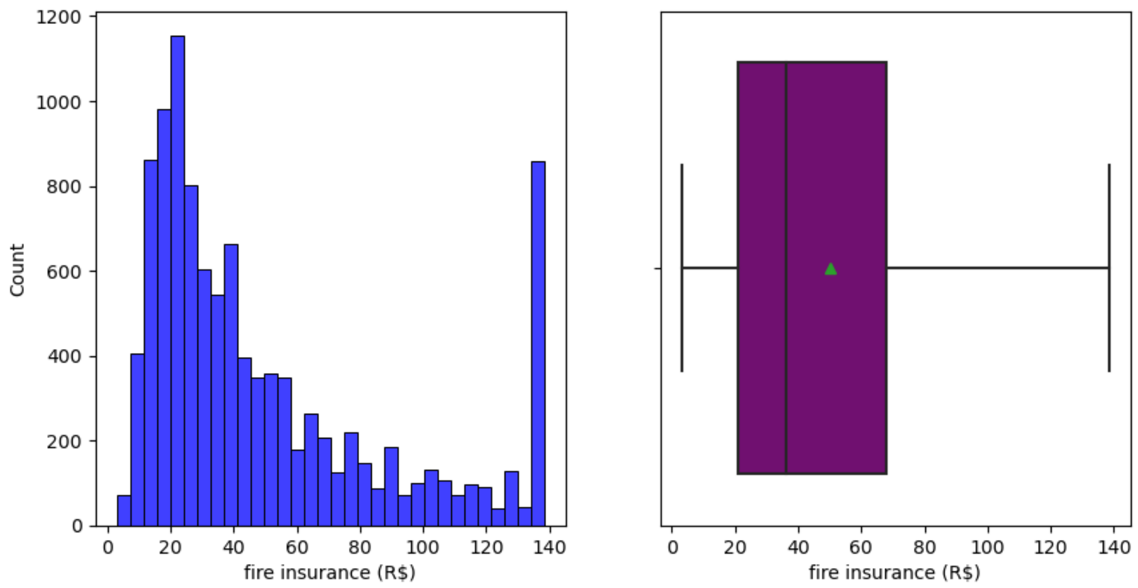
Interpretation:

Its observable that most people don't pay property tax and highest possible tax is 800 dollars

In [23]:

```
num_level1(empdf,numerical[4])
```

```
The mean of the fire insurance (R$) is 50.107510288065846
The median of the fire insurance (R$) is 36.0
The mode of the fire insurance (R$) is 138.5
The standard deviation of the fire insurance (R$) is 38.614564862056085
Number of missing values in the fire insurance (R$) is 0
```



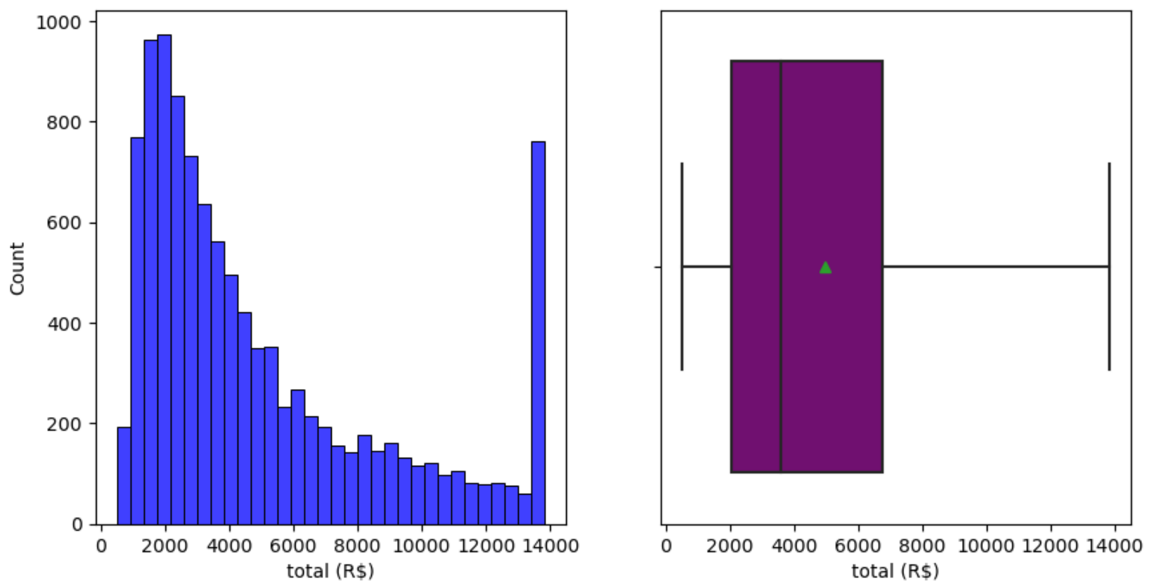
Interpretation:

we can see that the highest fire insurance ever claimed is 140 which is claimed also by considerably large people and many claims lie in range 10 to 60 dollars

In [24]:

```
num_level1(empdf,numerical[5])
```

```
The mean of the total (R$) is 4966.518308080808
The median of the total (R$) is 3581.5
The mode of the total (R$) is 13827.375
The standard deviation of the total (R$) is 3794.8994208776344
Number of missing values in the total (R$) is 0
```



Interpretation:

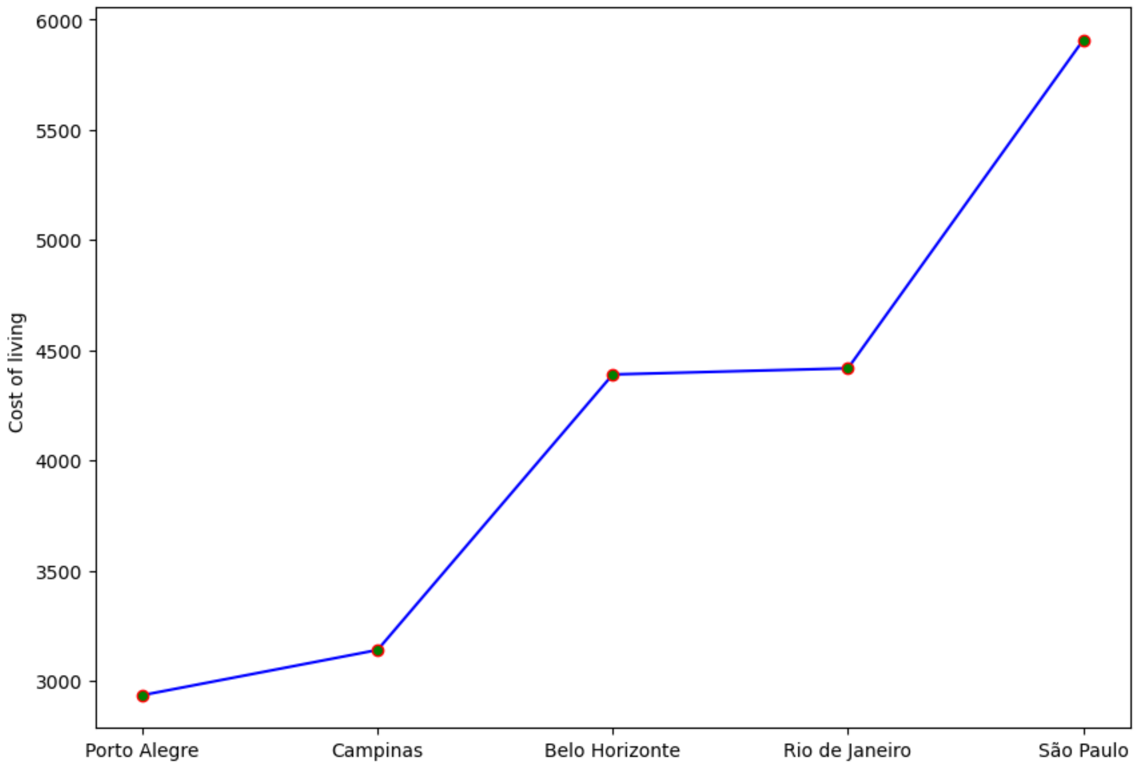
The highest cost ever is 14000 dollars for a home while most homes cost around 2500 to 5000 dollars

Level 2: Bivariate Analysis (Getting closer to the BIG QUESTION:)

Total cost of living vs City

In [25]:

```
fig, ax = plt.subplots(figsize = (10, 7))
city_col=empdf.groupby("city").mean()["total (R$)"].sort_values()
plt.plot(city_col, data = empdf,color="blue",marker="o", markeredgecolor="red", markerfacecolor="green")
plt.ylabel("Cost of living ")
plt.show()
```



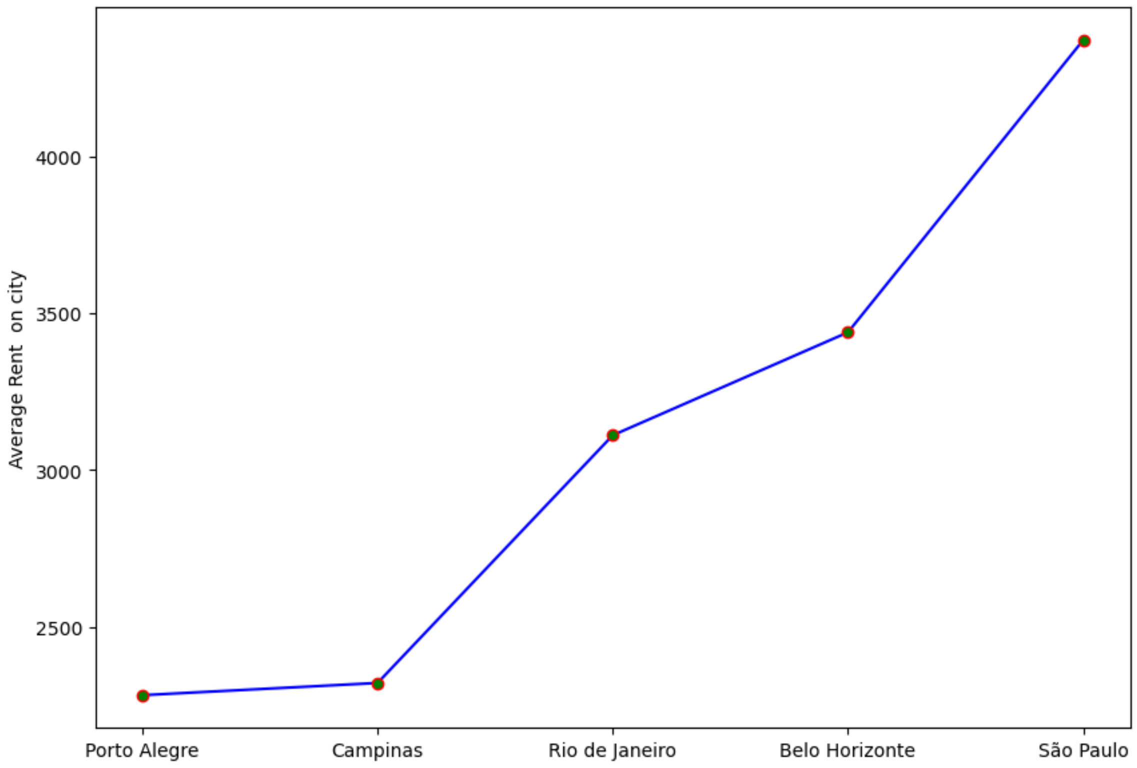
Interpretation:

From the above analysis we can see the cost of living is the highest in the city Sao Paulo

Total rent amount vs city

In [71]:

```
fig, ax = plt.subplots(figsize = (10, 7))
city_col=empdf.groupby("city").mean()["rent amount (R$)"].sort_values()
plt.plot(city_col, data = empdf,color="blue",marker="o", markeredgecolor="red", markerfacecolor="green")
plt.ylabel("Average Rent  on city")
plt.show()
```



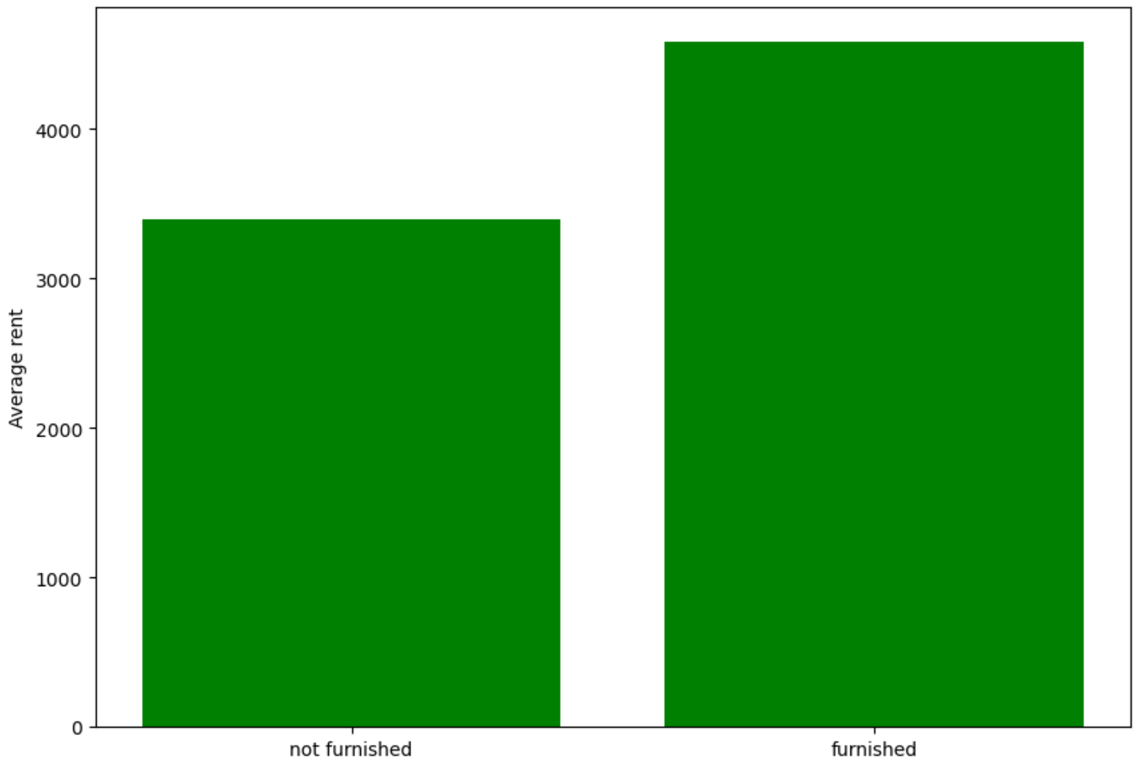
Interpretation:

We can see that again Sai Paulo is the city with highest cost of living

Furnishment vs average rent

In [38]:

```python
fig, ax = plt.subplots(figsize = (10, 7))
fur_col=empdf.groupby("furniture").mean()["rent amount (R$)"].sort_values()
plt.bar(fur_col.index,fur_col,color="green")
plt.ylabel("Average rent")
plt.show()
```



Interpretation:

We can see that amonst all homes furnished homes have higher rent than unfurnished homes

Level 3 - analysis

One could consider analyzing all the above columns for the customers who have left and having 2 or 3 dependents. However it could be a meaningless visualization, hence it is better to consult the domain expert to choose the appropriate columns for further analysis.

1. rental amount
2. property tax
3. rooms

In [39]:

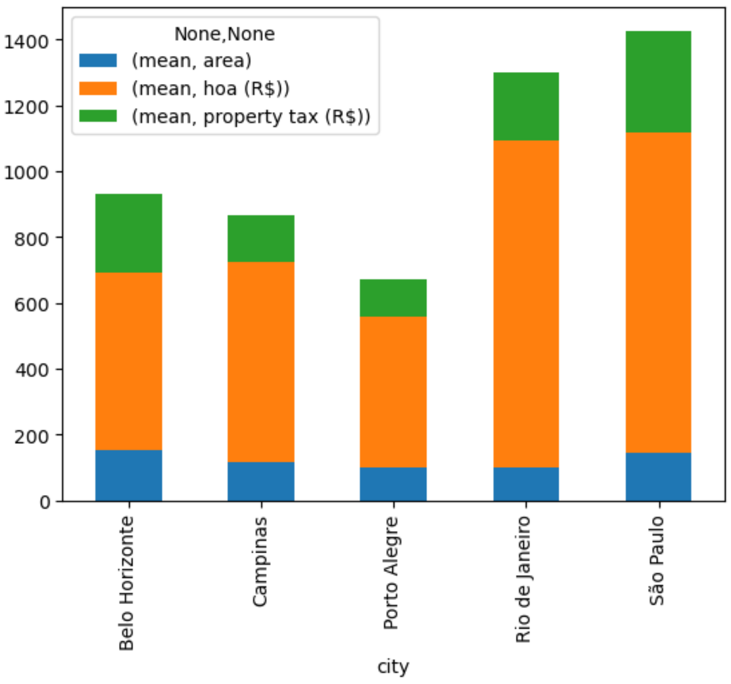
```python
print(tabulate({"Categorical":categorical,"continuous": numerical},headers = ["categorical", "numerical"]))
```

```
categorical     numerical
--------------  -------------------
city            area
rooms           hoa (R$)
bathroom        rent amount (R$)
parking spaces  property tax (R$)
floor           fire insurance (R$)
animal          total (R$)
furniture
```

In [62]:

```python
homes.plot(kind='bar', stacked=True)
```

Out[62]: `<AxesSubplot: xlabel='city'>`



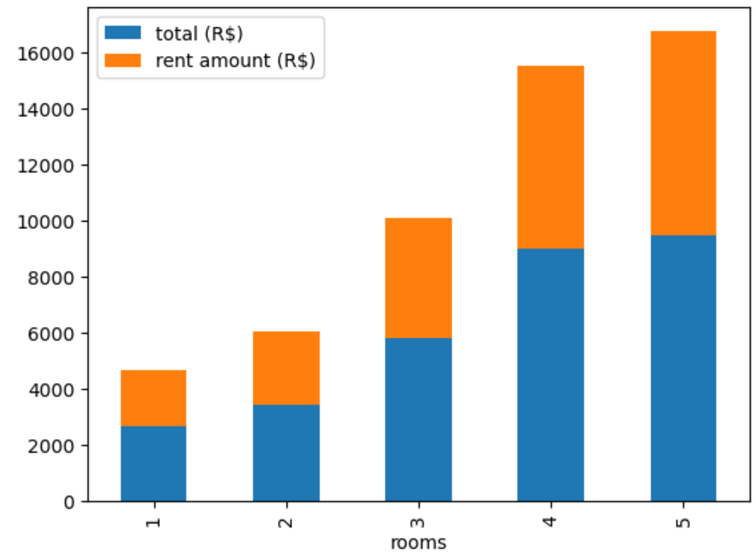
Interpretation :

We can see that on overall analysis of area and hoa and property tax Sao paulo and Rio de Janerio have the highest expense factor

Rooms vs total cost

In [82]:

```
empdf.groupby("rooms").mean().loc[:,["total (R$)","rent amount (R$)"]].plot(kind="bar",stacked=True)
plt.show()
```



Interpretations:

5&4 rooms have really high total cost of living compared to all other room levels