Task : A top real estate management firm wishes to help people choose an alternate city to relocate to.

```
As a data analyst, help the firm figure out suitable cities for relocation
for bachelors, for mid-sized families and for large families.
```

In [2]:

```python
# importing the important libraries

import matplotlib.pyplot as plt             # to visualize
from tabulate import tabulate               # to print the table
import matplotlib as mat                    # to visualize
import seaborn as sns                       # to visualize
import pandas as pd                         # for data reading
import numpy as np
```

In [3]:

```python
df = pd.read_csv("DS1_C5_S3_BazilHousing_Data_Hackathon.csv")
df.sample(7)
```

Out[3]:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)
7468	São Paulo	220	3	4	2	0	acept	furnished	0	7000	434
5534	São Paulo	48	1	1	1	4	acept	furnished	1450	4200	135
6085	Belo Horizonte	32	1	1	0	9	not acept	not furnished	300	1000	100
8482	São Paulo	115	4	4	3	2	acept	not furnished	2600	4000	417
3846	São Paulo	400	3	4	4	0	acept	not furnished	0	12000	1667
10539	Campinas	144	3	2	3	0	acept	not furnished	890	4000	138
2311	Rio de Janeiro	100	2	2	1	6	acept	furnished	2400	8900	380

In [4]:

```
df.isnull().sum()       # isnull returns the True/False dataframe
                            # sum: counts the number of True in columns
```

Out[4]:

```
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
```

There are no missing values in the dataframe so we can start our analysis.

In [5]:

```
# As mentioned, The cities 'Rio de Janeiro' and 'Sao Paulo' are very expensive,
# So,let's separate the data for less expensive cities only

df1 = df.loc[(df['city'].isin(['Porto Alegre','Campinas','Belo Horizonte']))]
df1
```

Out[5]:

	city	area	rooms	bathroom	parking spaces	floor	animal	furniture	hoa (R$)	rent amount (R$)	property tax (R$)	ir
2	Porto Alegre	80	1	1	1	6	acept	not furnished	1000	2800	0	
3	Porto Alegre	51	2	1	0	2	acept	not furnished	270	1112	22	
11	Campinas	46	1	1	1	10	acept	not furnished	550	580	43	
15	Campinas	330	4	6	6	0	acept	furnished	680	8000	328	
21	Belo Horizonte	42	1	1	1	17	not acept	furnished	470	2690	172	
...	...	...	...	...	...	...	...	...	...	...	...	
10667	Belo Horizonte	75	2	1	1	3	not acept	not furnished	180	1250	0	
10673	Porto Alegre	220	3	2	2	15	acept	not furnished	842	2400	117	
10676	Porto Alegre	40	1	1	0	1	acept	not furnished	330	1200	159	
10682	Porto Alegre	160	3	2	3	4	acept	furnished	850	3300	220	
10687	Porto Alegre	63	2	1	1	5	not acept	furnished	402	1478	24	

3304 rows × 13 columns

In [6]:

```
# Seprating out the categorical and continuous variables
def seprate_data_types(df):
    categorical = []
    continuous = []
    for column in df.columns:                # Looping on the number of column
        if df[column].dtype == object:

            categorical.append(column)
        else:
            continuous.append(column)

    return categorical, continuous


categorical, continuous = seprate_data_types(df)         # Calling the functio

# # Tabulate is a package used to print the list, dict or any data sets in a pr
from tabulate import tabulate
table = [categorical, continuous]
print(tabulate({"Categorical":categorical,
                "continuous": continuous}, headers = ["categorical", "continuou
```

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

Analysis for Bachelors :

For bachelors, we'll consider the following criteria:

- 2 or less than 2 rooms
- 1 bathroom
- rent should be less than 2000
- require both furnished and non furnished

In [7]:

```
# filter the data based on the given criteria
bachelors = df1[(df1['rooms'] <= 2) & (df1['bathroom'] == 1) & (df1['rent amou
```

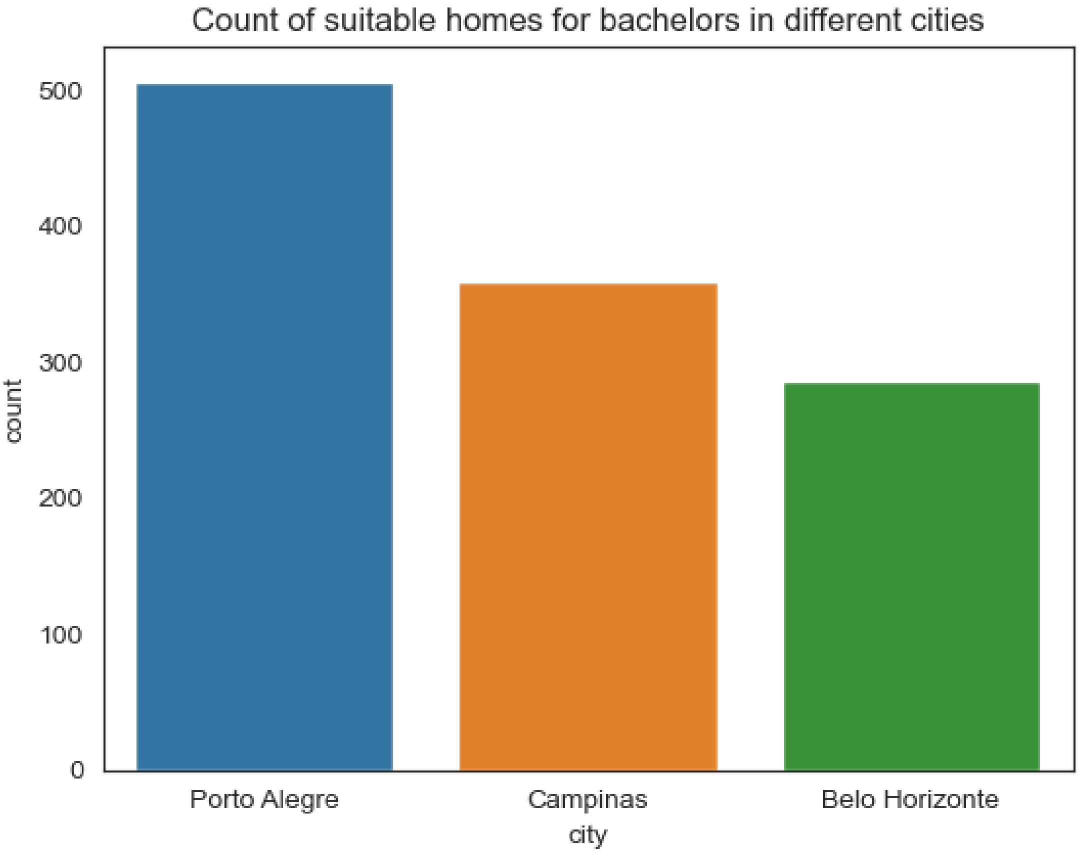
In [8]:

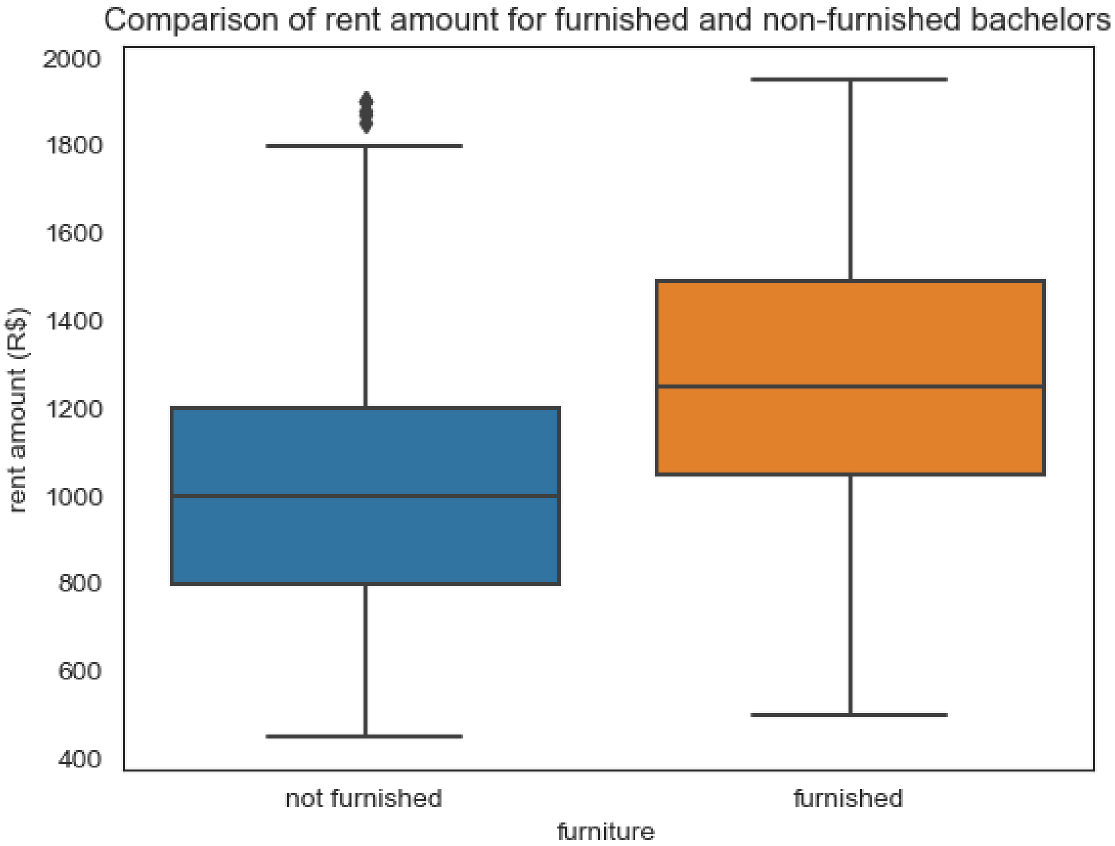
```
# check the shape of the filtered data
print("Shape of bachelors data:", bachelors.shape)
```

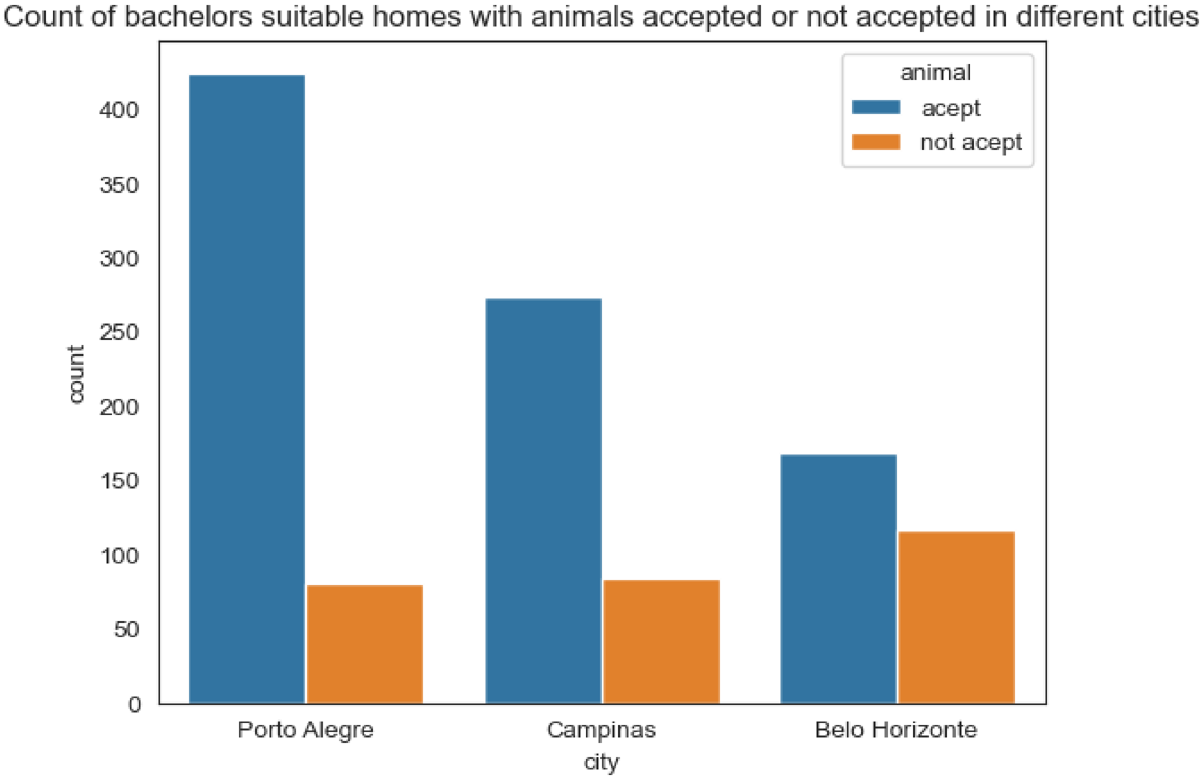
```
Shape of bachelors data: (1151, 13)
```

In [11]:

```
# explore the data using different plots and graphs

sns.countplot(x='city', data=bachelors)
plt.title('Count of suitable homes for bachelors in different cities')
plt.show()

sns.boxplot(x='furniture', y='rent amount (R$)', data=bachelors)
plt.title('Comparison of rent amount for furnished and non-furnished bachelors
plt.show()

acept_animal = bachelors[bachelors['animal'] == 'acept']
not_acept_animal = bachelors[bachelors['animal'] == 'not acept']

# Create a count plot showing the number of suitable homes with animals accepte
sns.countplot(x='city', hue='animal', data=bachelors)
plt.title('Count of bachelors suitable homes with animals accepted or not accep
plt.show()
```







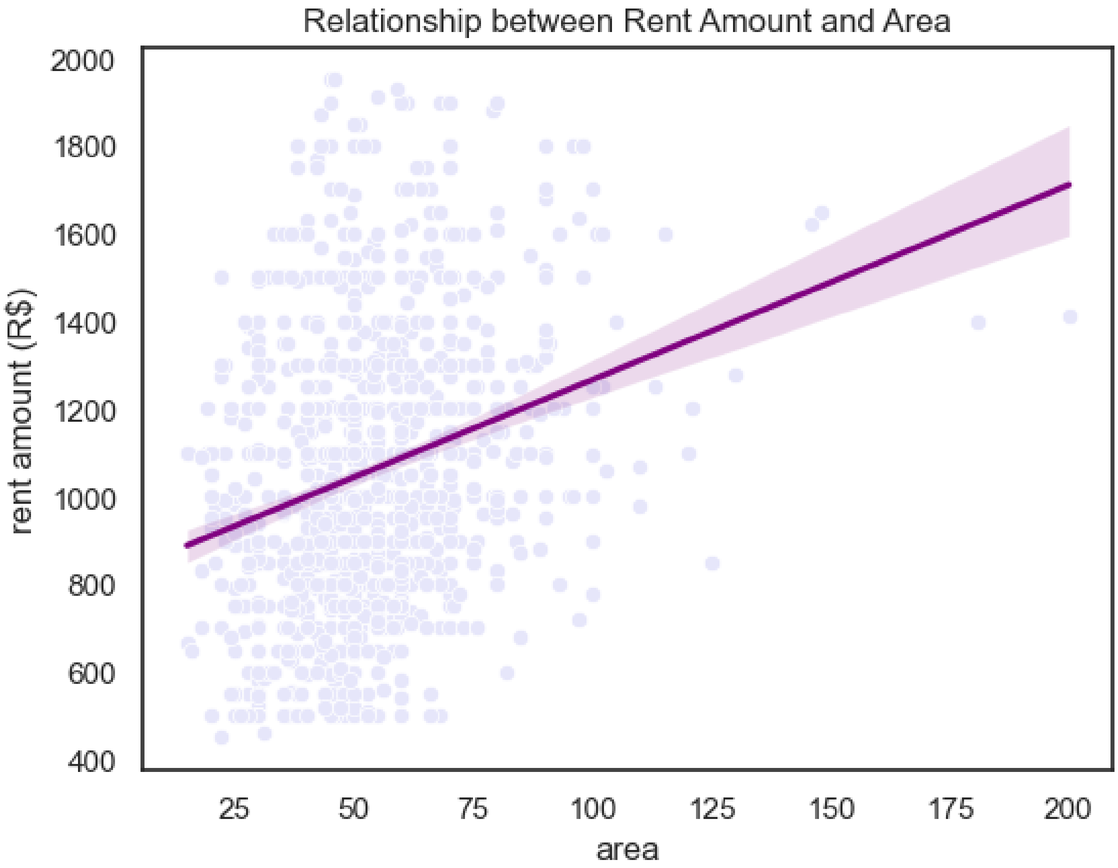
Interpretations :

- Graph 1 : The count plot is showing the number of suitable bachelor homes available in different cities based on the given criteria. The plot shows that Porto Alegre has the highest number of suitable homes for bachelors, followed by Campinas. Belo Horizonte has the lowest number of suitable homes for bachelors based on the given criteria.

- Graph 2 : The boxplot is comparing the rent amount for furnished and non-furnished homes suitable for bachelors. The plot shows that non-furnished homes have a lower median rent compared to furnished homes. However, there is a considerable overlap between the two categories, indicating that there are furnished homes available for bachelors with similar rent amounts as non-furnished homes.
- Graph 3 : The plot shows that, based on the given criteria, there are more suitable homes available for bachelors with animals accepted in all three cities - Porto Alegre, Campinas, and Belo Horizonte. However, in 'Porto Alegre' the number of suitable homes with animals accepted is almost triple the number of suitable homes with animals not accepted. This plot can be useful for bachelors who are looking for homes with animals accepted or not accepted, depending on their preferences.

In [41]:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.scatterplot(x='area', y='rent amount (R$)', data=bachelors, color='lavende
sns.regplot(x='area', y='rent amount (R$)', data=bachelors, scatter=False, col
plt.title('Relationship between Rent Amount and Area')
plt.show()
```



Interpretation :

- we can see that there is a positive linear relationship between area and rent amount (R$). In other words, as the area of the bachelor apartments increases, the rent amount tends to increase as well. However, the scatter of data points around the trend line indicates that the relationship is not perfect and there may be some variation in rent amount for a given area.

Overall Interpretation:

Based on the given criteria, the majority of the suitable homes for bachelors are located in Porto Alegre followed by Campinas.

Analysis for Mid-sized Families :

For Mid-sized Families, we'll consider the following criteria:

- more than 2 rooms
- more than 2 bathrooms
- more than 1 parking spaces
- rent should be less than 5000
- furnished or not furnished both
- animal accepted or not accepted both
- floor more than 2

In [15]:

```
# filter the data based on the given criteria
mid_fam = df1[(df1['rooms'] > 2) & (df1['bathroom'] > 1) & (df1['floor'] > 1) &
```

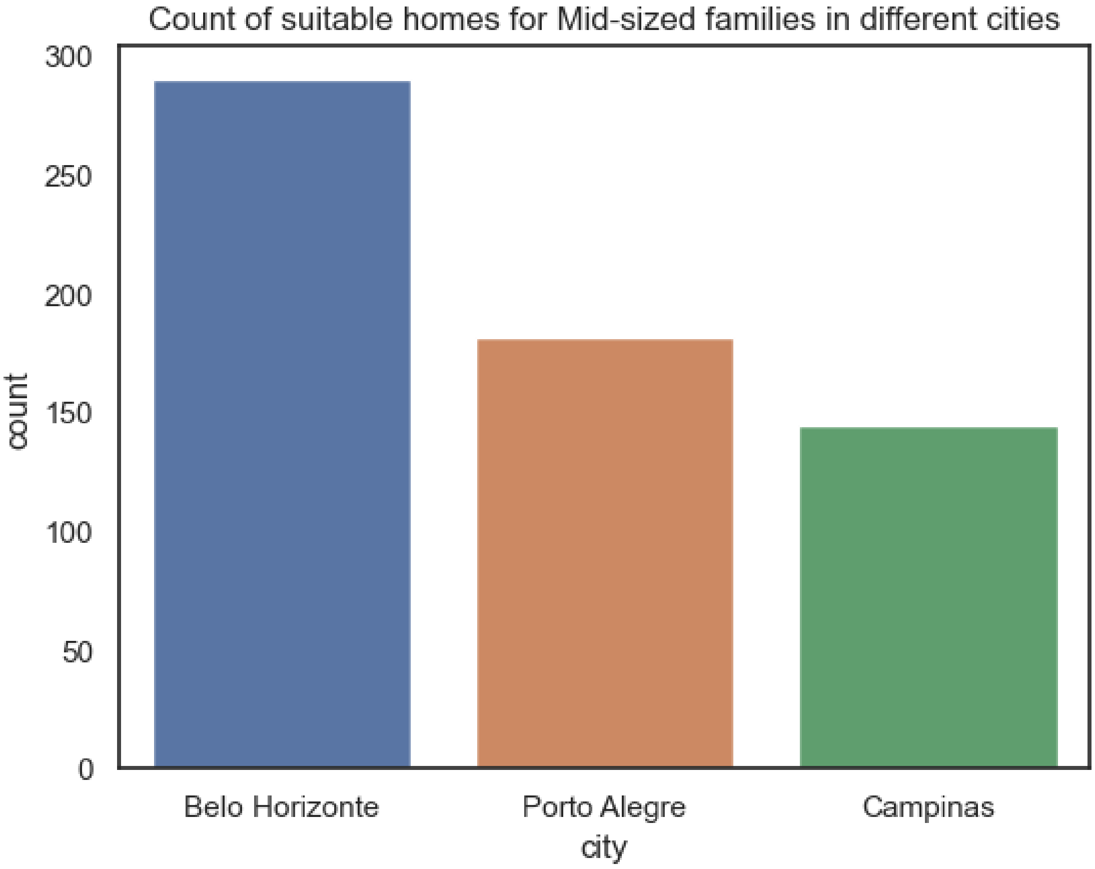
In [16]:

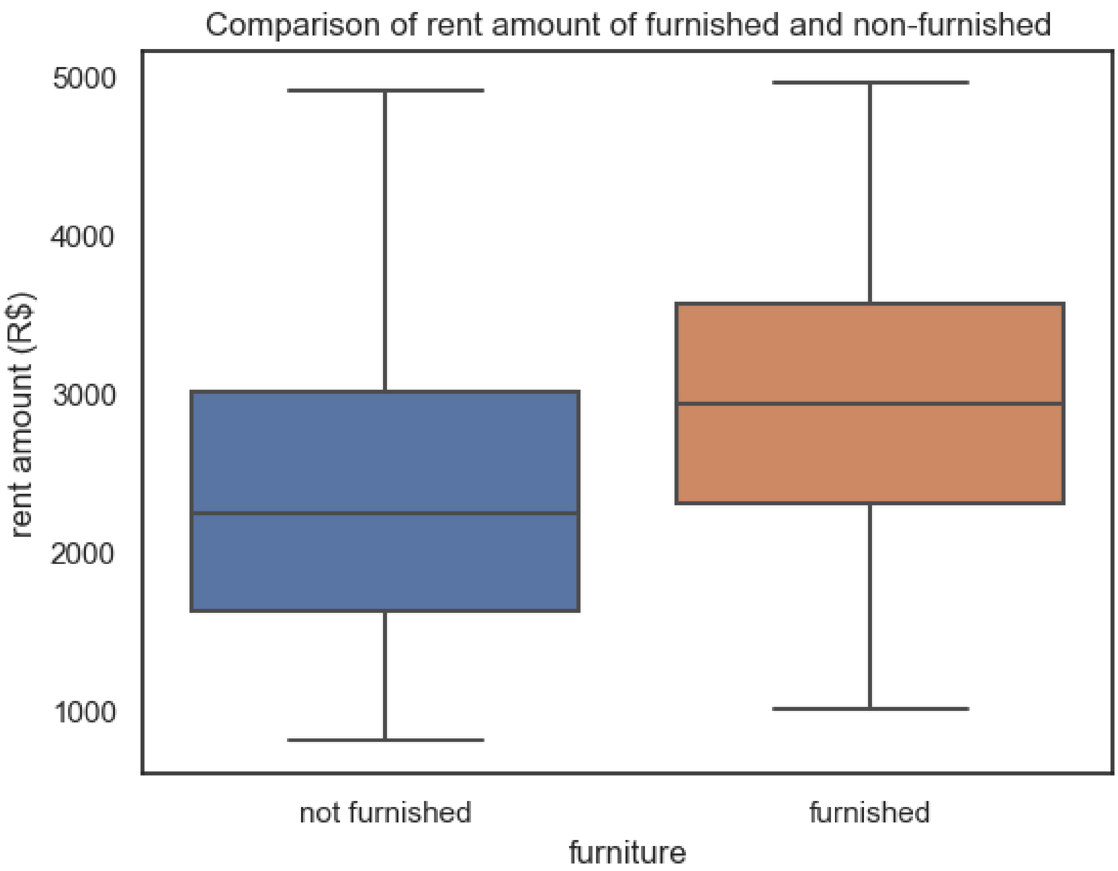
```
# check the shape of the filtered data
print("Shape of bachelors data:", mid_fam.shape)
```

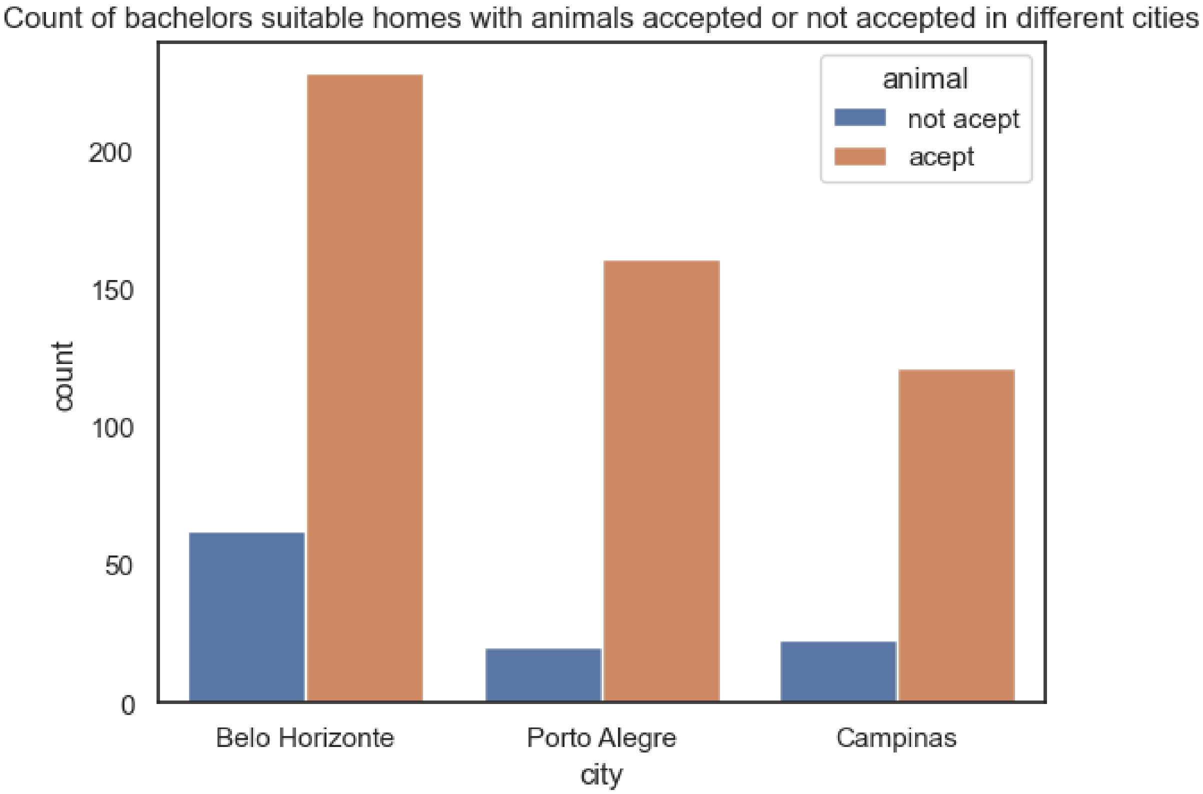
```
Shape of bachelors data: (615, 13)
```

In [43]:

```
# explore the data using different plots and graphs

sns.countplot(x='city', data=mid_fam)
plt.title('Count of suitable homes for Mid-sized families in different cities'
plt.show()

sns.boxplot(x='furniture', y='rent amount (R$)', data=mid_fam)
plt.title('Comparison of rent amount of furnished and non-furnished')
plt.show()

acept_animal = mid_fam[mid_fam['animal'] == 'acept']
not_acept_animal = mid_fam[mid_fam['animal'] == 'not acept']

sns.countplot(x='city', hue='animal', data=mid_fam)
plt.title('Count of bachelors suitable homes with animals accepted or not accep
plt.show()
```







Interpretations :

- Graph 1 : The Count plot shows that 'Belo Horizonte' has the highest number of suitable homes for mid sized families, followed by Porto Alegre and Campinas based on the given criteria.

- Graph 2 : The Box plot shows that furnished houses have relatively higher house rent than non furnished ones.

- Graph 3 : In 'Belo Horizonte' the number of suitable homes with animals accepted is much more

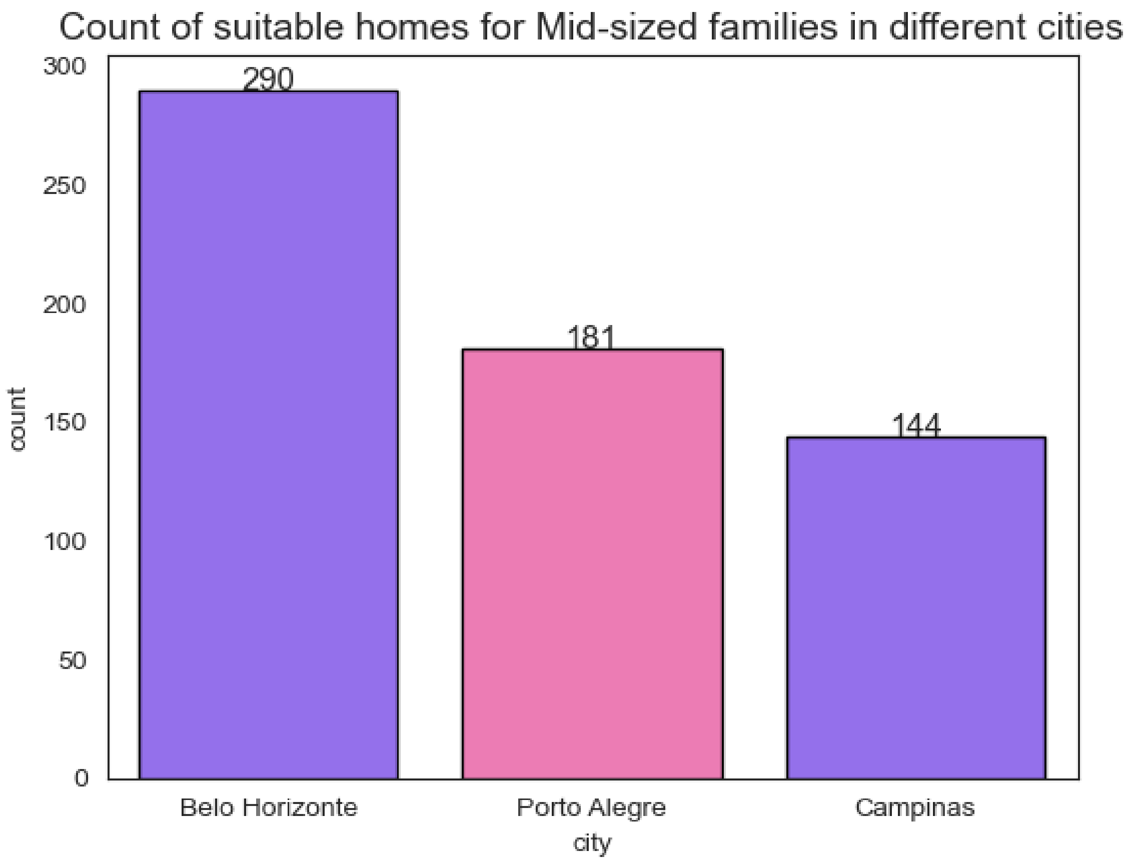
Overall Interpretation:

Based on the given criteria, the majority of the suitable homes for mid-sized families are located in 'Belo Horizonte'.

Customizing the above graphs

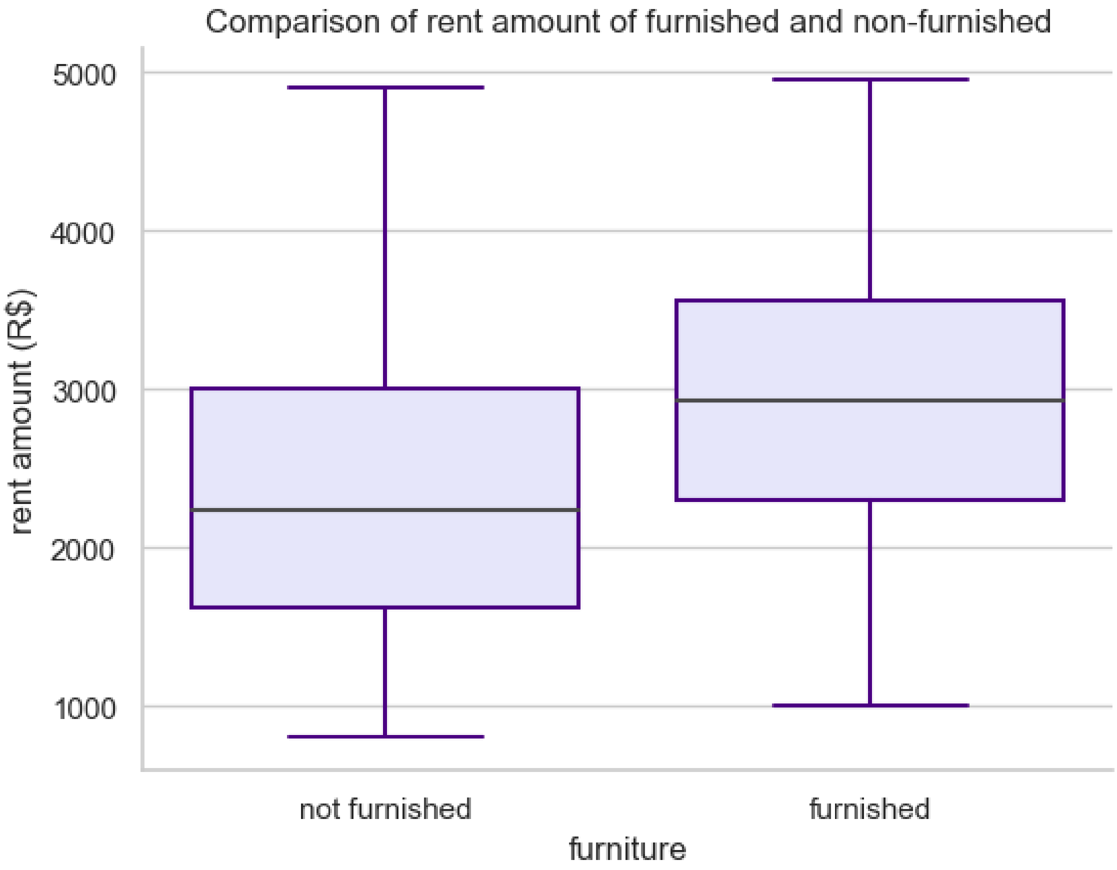
In [18]:

```python
import seaborn as sns
import matplotlib.pyplot as plt

# Set color palette
colors = ['#8B5BFF', '#FF69B4']

# Create countplot
ax = sns.countplot(x='city', data=mid_fam, palette=colors, edgecolor='black')

# Set title
ax.set_title('Count of suitable homes for Mid-sized families in different citie

# Add count labels on bars
for p in ax.patches:
    ax.annotate(f'\n{p.get_height()}', (p.get_x()+0.4, p.get_height()), ha='ce

# Show plot
plt.show()
```



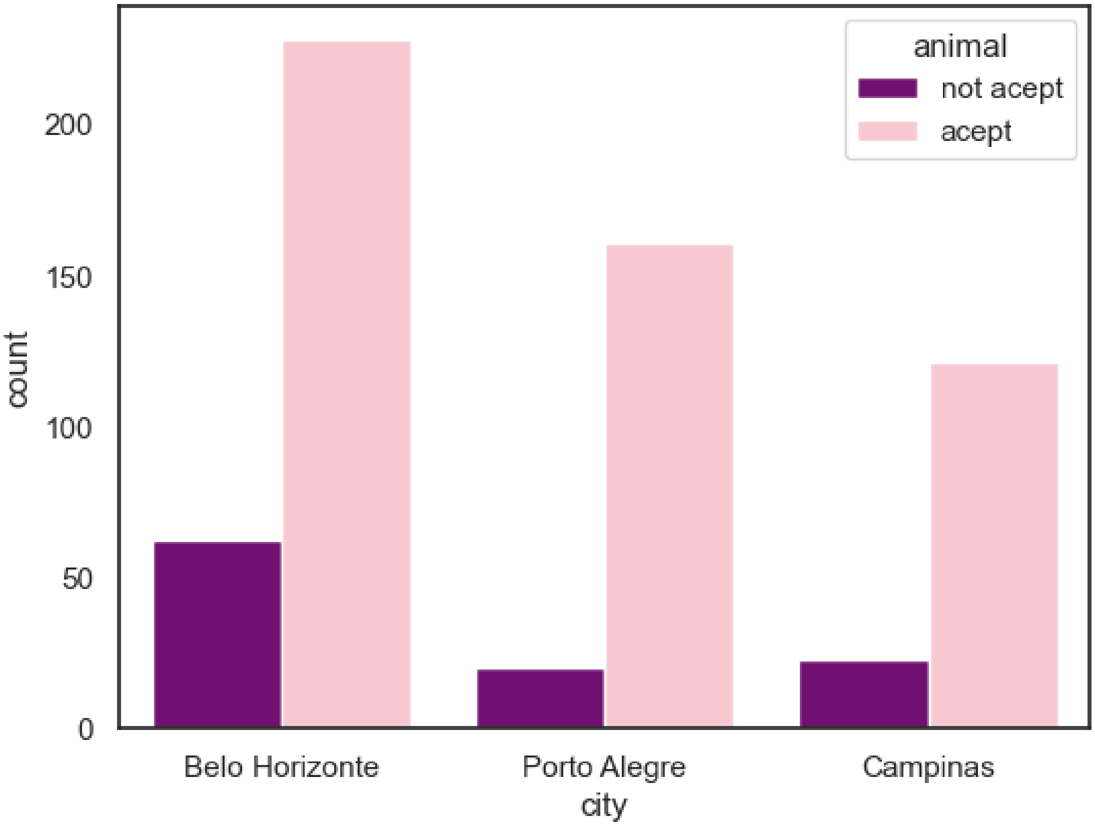
In [19]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# assume that mid_fam is a DataFrame containing rental information
# for mid-sized families
sns.set(style='whitegrid')
purple = '#4b0082'
lavender = '#e6e6fa'
sns.boxplot(x='furniture', y='rent amount (R$)', data=mid_fam,
            boxprops=dict(edgecolor=purple, facecolor=lavender),
            whiskerprops=dict(color=purple),
            capprops=dict(color=purple))
plt.title('Comparison of rent amount of furnished and non-furnished')
sns.despine()
plt.show()
```



In [21]:
```
sns.countplot(x='city', hue='animal', data=mid_fam,
              palette={'acept': 'pink', 'not acept': 'purple'})
```

Out[21]:
```
<AxesSubplot:xlabel='city', ylabel='count'>
```



Analysis for Large-sized Families :

For Large-sized Families, we'll consider the following criteria:

- more than 2 rooms
- more than 2 bathrooms
- more than 2 parking spaces
- rent should be less than 9000
- furnished or not furnished both
- animal accepted or not accepted both
- floor more than 2

In [22]:
```
# filter the data based on the given criteria
large_fam = df1[(df1['rooms'] > 2) & (df1['bathroom'] > 2) & (df1['floor'] > 1
```

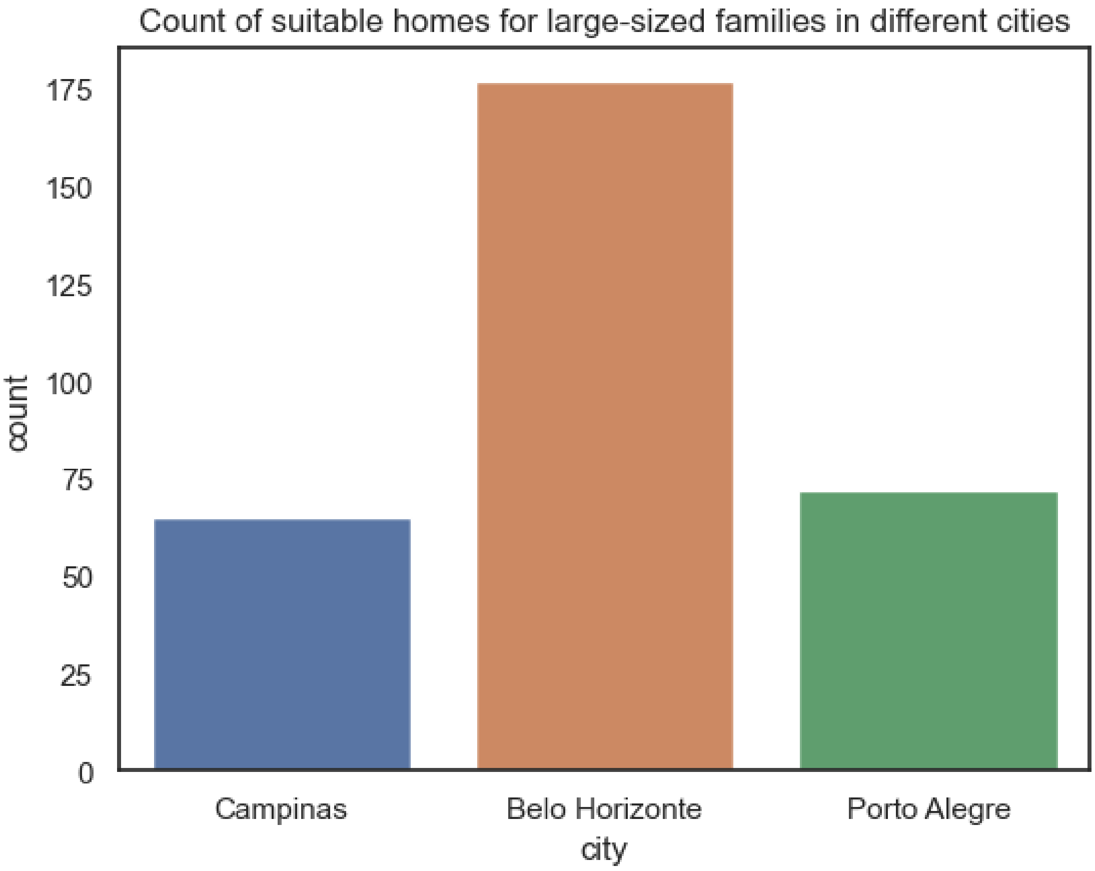
In [23]:

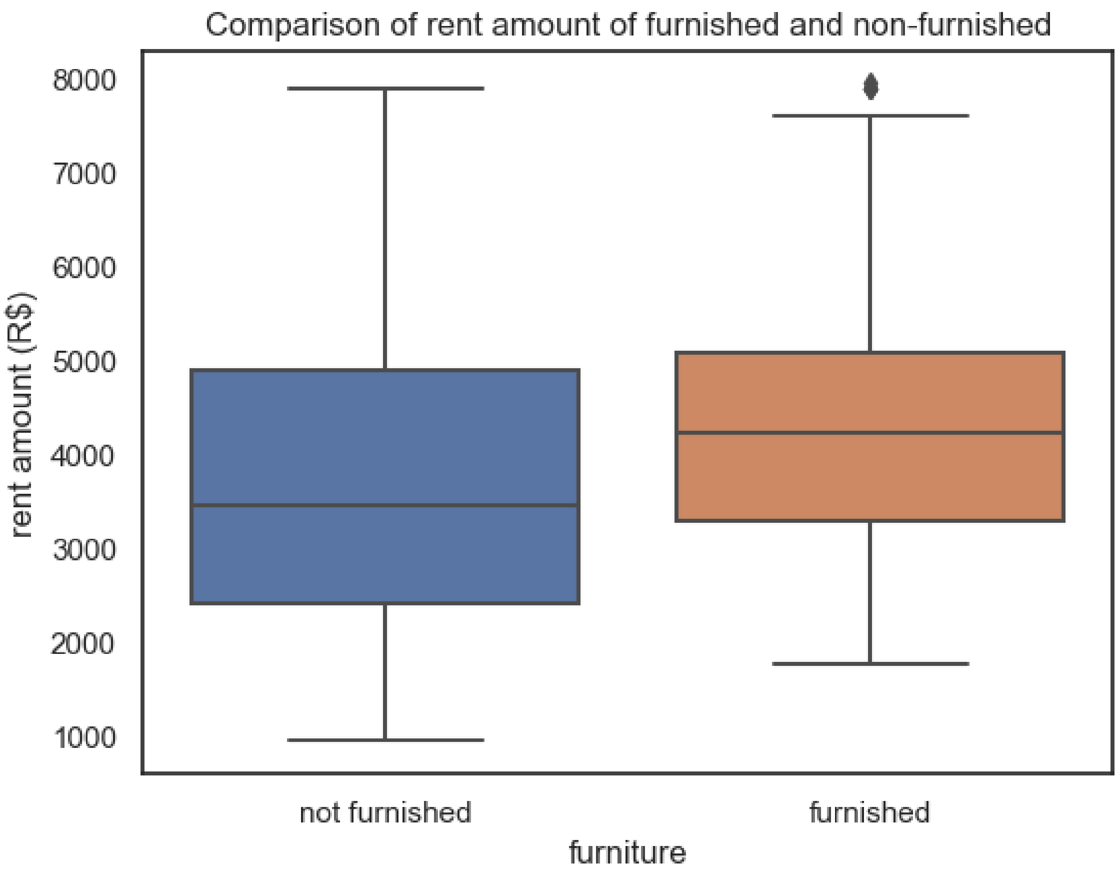
```
# check the shape of the filtered data
print("Shape of bachelors data:", large_fam.shape)
```

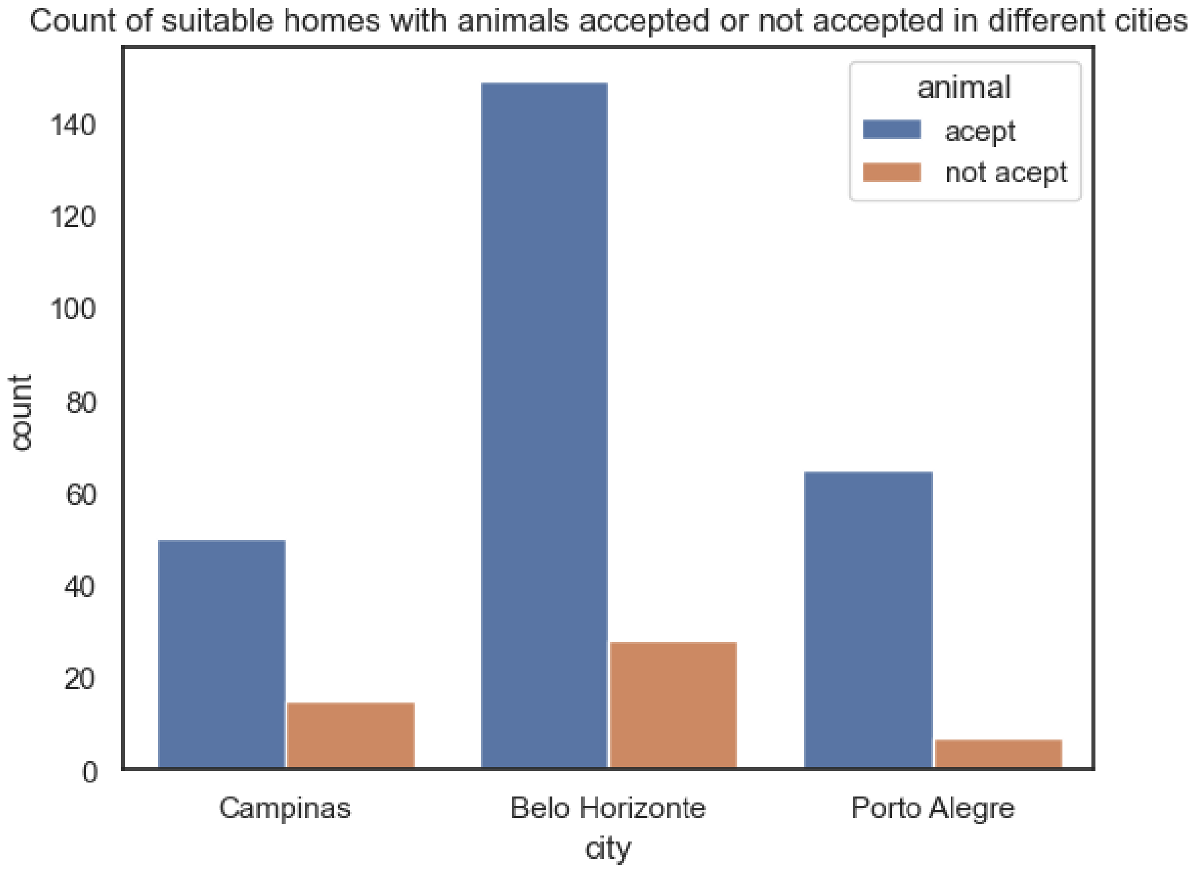
```
Shape of bachelors data: (314, 13)
```

In [24]:

```
# explore the data using different plots and graphs

sns.countplot(x='city', data=large_fam)
plt.title('Count of suitable homes for large-sized families in different citie
plt.show()

sns.boxplot(x='furniture', y='rent amount (R$)', data=large_fam)
plt.title('Comparison of rent amount of furnished and non-furnished')
plt.show()

acept_animal = large_fam[large_fam['animal'] == 'acept']
not_acept_animal = large_fam[large_fam['animal'] == 'not acept']

sns.countplot(x='city', hue='animal', data=large_fam)
plt.title('Count of suitable homes with animals accepted or not accepted in di
plt.show()
```





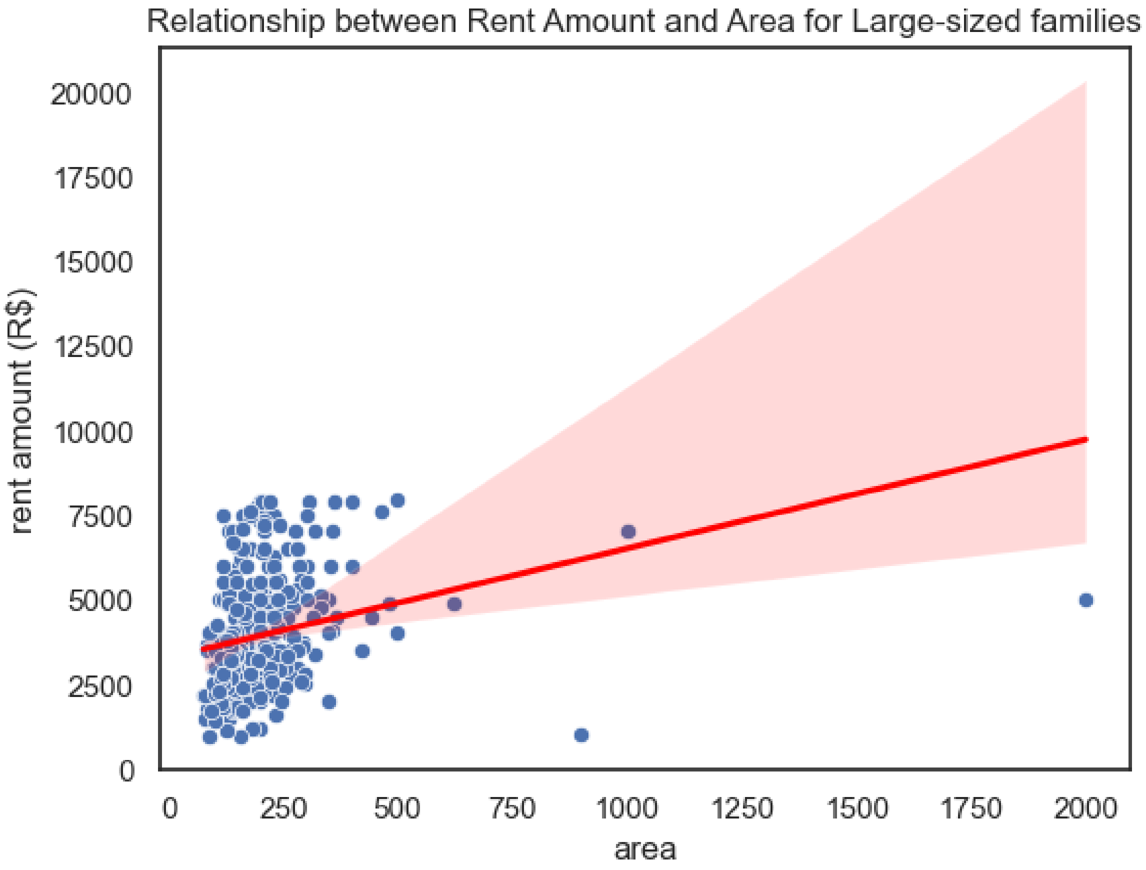


Interpretations :

- Graph 1 : The Count plot shows that 'Belo Horizonte' has the highest number of suitable homes for large sized families, followed by Porto Alegre and Campinas based on the given criteria.
- Graph 2 : The Box plot shows that furnished houses have relatively higher house rent than non-furnished ones.
- Graph 3 : In 'Belo Horizonte' the number of suitable homes with animals accepted is much more than the number of suitable homes with animals not accepted.

In [25]:

```
sns.scatterplot(x='area', y='rent amount (R$)', data=large_fam)
sns.regplot(x='area', y='rent amount (R$)', data=large_fam, scatter=False, colo
plt.title('Relationship between Rent Amount and Area for Large-sized families'
plt.show()
```



Interpretation :

The scatter plot indicates that there is a positive correlation between the area of the homes and the rent amount. As the area of the homes increases, the rent amount also tends to increase. However, there are some outliers where the rent amount is higher than what would be expected based on the area of the home.

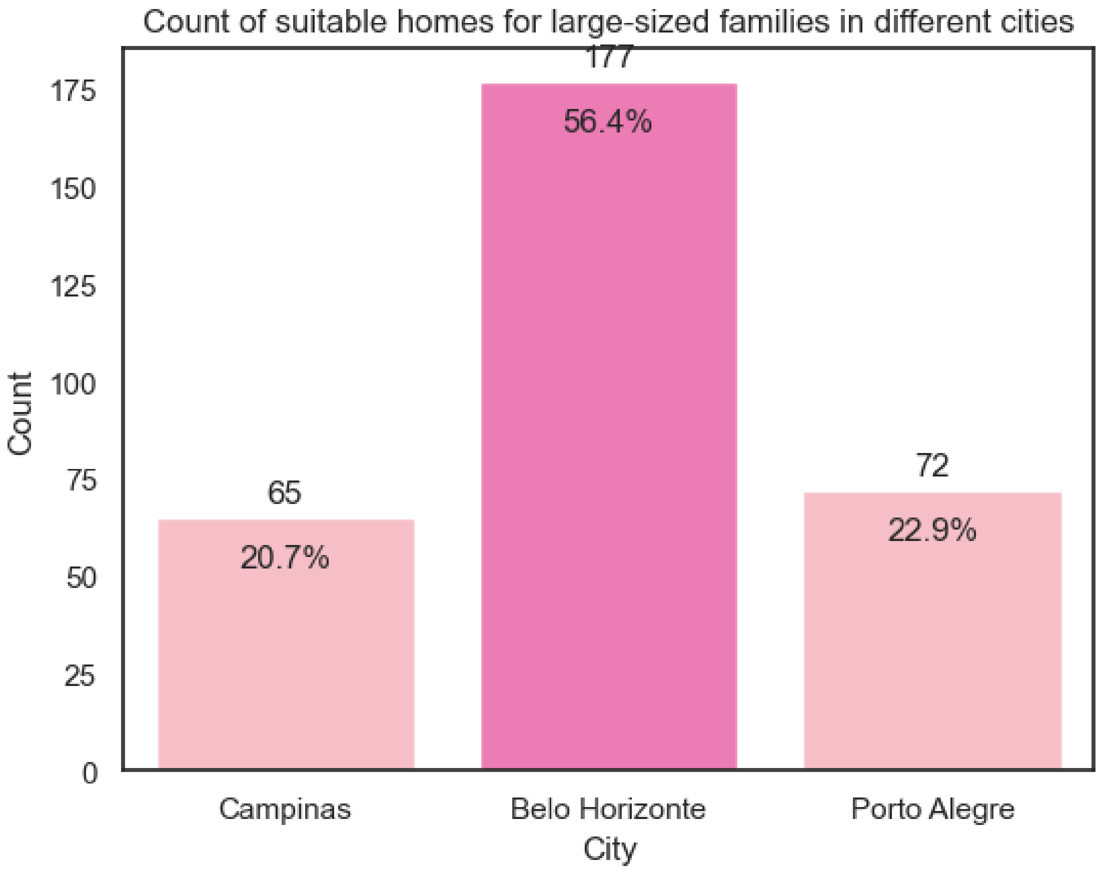
Overall Interpretation:

Based on the given criteria, the majority of the suitable homes for large-sized families are located in

Customizing the above graphs

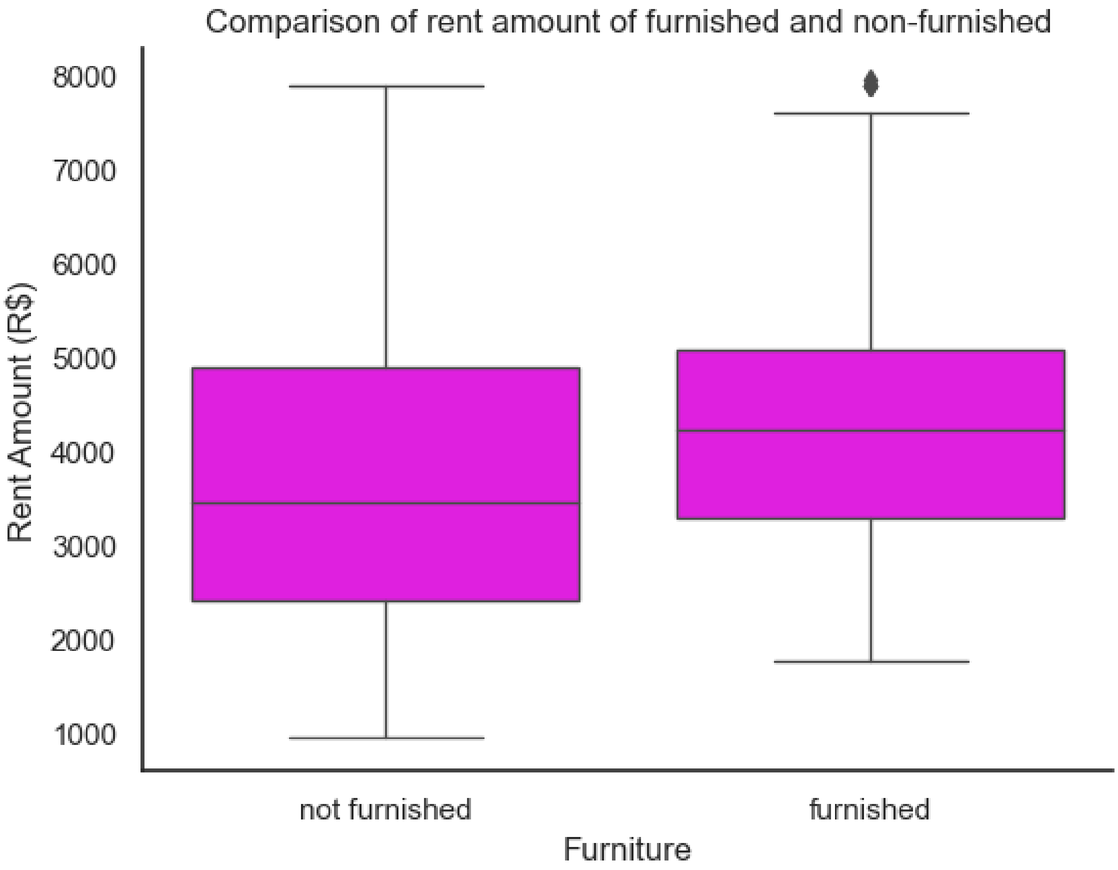
In [27]:

```
import seaborn as sns

sns.set_style('white')
ax = sns.countplot(x='city', data=large_fam, palette=['#FFB6C1', '#FF69B4'])
plt.title('Count of suitable homes for large-sized families in different citie
plt.xlabel('City')
plt.ylabel('Count')
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, 9),
                textcoords = 'offset points')
    ax.annotate('{:.1%}'.format(p.get_height()/len(large_fam)),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, -15),
                textcoords = 'offset points')
plt.show()

```



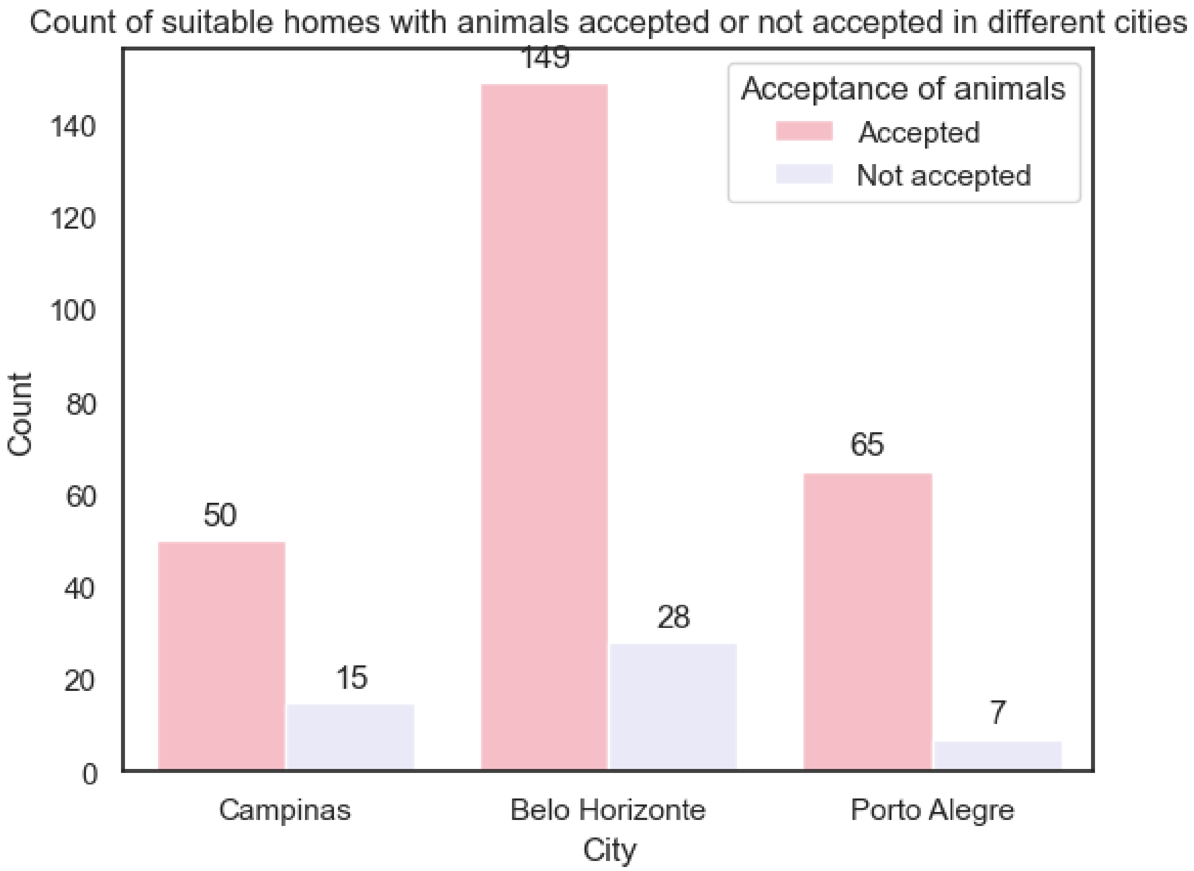
In [37]:

```
import seaborn as sns

sns.set_style('white')
sns.boxplot(x='furniture', y='rent amount (R$)', data=large_fam, color='magent
sns.despine()
plt.title('Comparison of rent amount of furnished and non-furnished')
plt.xlabel('Furniture')
plt.ylabel('Rent Amount (R$)')
plt.show()

```



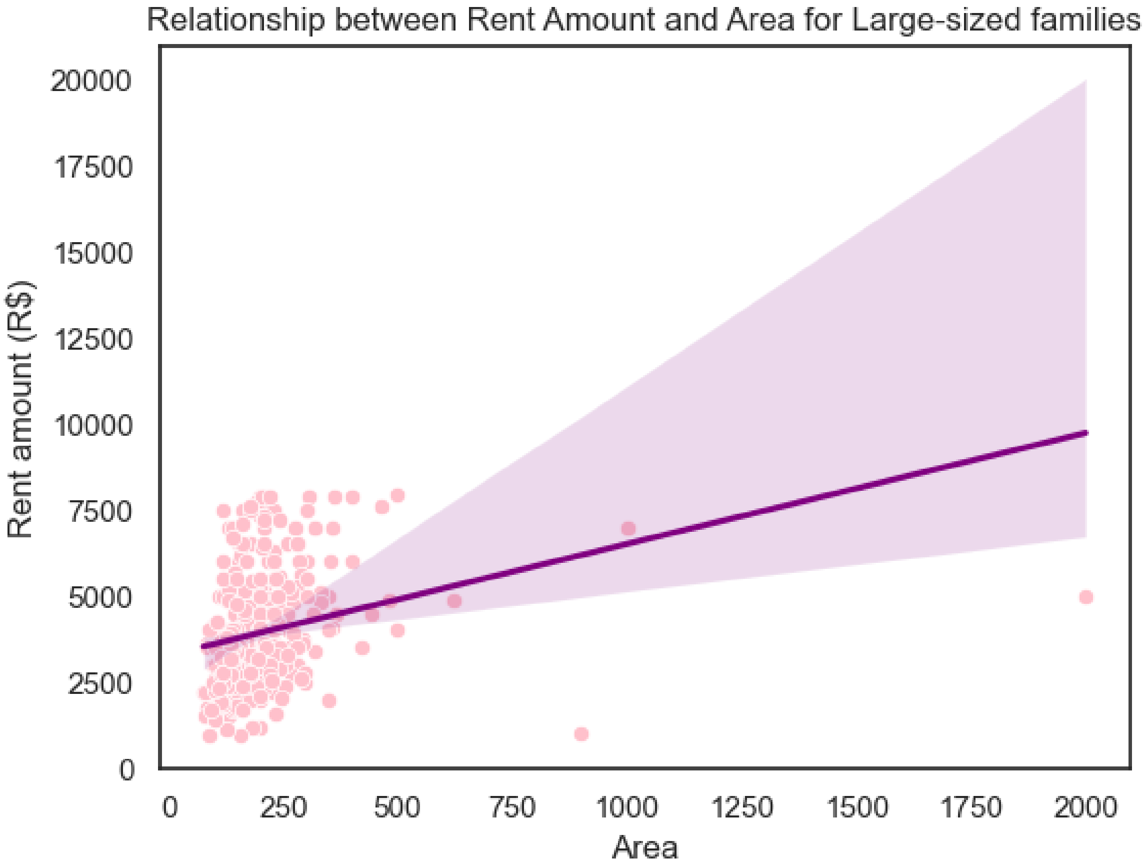
In [38]:

```
import seaborn as sns

acept_animal = large_fam[large_fam['animal'] == 'acept']
not_acept_animal = large_fam[large_fam['animal'] == 'not acept']

sns.set_style('white')
ax = sns.countplot(x='city', hue='animal', data=large_fam, palette=['#FFB6C1',
plt.title('Count of suitable homes with animals accepted or not accepted in di
plt.xlabel('City')
plt.ylabel('Count')
for p in ax.patches:
    ax.annotate(format(p.get_height(), '.0f'),
                (p.get_x() + p.get_width() / 2., p.get_height()),
                ha = 'center', va = 'center',
                xytext = (0, 9),
                textcoords = 'offset points')
plt.legend(title='Acceptance of animals', labels=['Accepted', 'Not accepted'])
plt.show()

```



In [40]:

```
import seaborn as sns

sns.set_style('white')
sns.scatterplot(x='area', y='rent amount (R$)', data=large_fam, color='pink')
sns.regplot(x='area', y='rent amount (R$)', data=large_fam, scatter=False, col
plt.title('Relationship between Rent Amount and Area for Large-sized families'
plt.xlabel('Area')
plt.ylabel('Rent amount (R$)')
plt.show()

```


Relationship between Rent Amount and Area for Large-sized families
Rent amount (R$)
Area