

Relatório de Análise VII

Criando Agrupamentos

In [1]: `# importando Pandas
import pandas as pd`

In [2]: `# importando a base de dados
dados = pd.read_csv('data/aluguel_residencial.csv', sep = ';')`

In [3]: `# exibindo os primeiros dez itens
dados.head(10)`

Out[3]:

	Tipo	Bairro	Quartos	Vagas	Suites	Area	Valor	Condominio	IPTU	Valor m2	Tipo Agregado
0	Quitinete	Copacabana	1	0	0	40	1700.0	500.0	60.0	42.50	Apartamento
1	Casa	Jardim Botânico	2	0	1	100	7000.0	0.0	0.0	70.00	Casa
2	Apartamento	Centro	1	0	0	15	800.0	390.0	20.0	53.33	Apartamento
3	Apartamento	Higienópolis	1	0	0	48	800.0	230.0	0.0	16.67	Apartamento
4	Apartamento	Cachambi	2	0	0	50	1300.0	301.0	17.0	26.00	Apartamento
5	Casa de Condomínio	Barra da Tijuca	5	4	5	750	22000.0	0.0	0.0	29.33	Casa
6	Casa de Condomínio	Ramos	2	2	0	65	1000.0	0.0	0.0	15.38	Casa
7	Apartamento	Grajaú	2	1	0	70	1500.0	642.0	74.0	21.43	Apartamento
8	Apartamento	Lins de Vasconcelos	3	1	1	90	1500.0	455.0	14.0	16.67	Apartamento
9	Apartamento	Copacabana	1	0	1	40	2000.0	561.0	50.0	50.00	Apartamento

In [4]: `# obtendo a média geral dos valores
dados['Valor'].mean()`

Out[4]: 5046.172821405663

In [5]: `# coletando a média segundo alguns tipos de variáveis
para facilitar a visualização, a seleção contém apenas alguns bairros
bairros = ['Barra da Tijuca', 'Copacabana', 'Ipanema', 'Leblon', 'Botafogo', 'Flamengo', 'Tijuca']
selecao = dados['Bairro'].isin(bairros)
dados = dados[selecao]`

In [6]: `# utilizando o método drop_duplicates()
para remover as duplicatas em nossa listagem
dados['Bairro'].drop_duplicates()`

Out[6]:

```
0      Copacabana
5      Barra da Tijuca
15     Tijuca
19     Ipanema
23     Botafogo
27     Leblon
78     Flamengo
Name: Bairro, dtype: object
```

In [7]: `# criando nossos primeiros grupos com base dos bairros selecionados
grupo_bairro = dados.groupby('Bairro')`

In [8]: `# verificando o tipo da variável
type(grupo_bairro)`

Out[8]: pandas.core.groupby.generic.DataFrameGroupBy

In [9]: `# propriedade groups, que consiste em um dicionário em que a
chave é variável utilizada como indexador e os índices em
que os registros de bairro foram encontrados
grupo_bairro.groups`

Out[9]:

```
{'Barra da Tijuca': Int64Index([ 5, 14, 16, 21, 30, 32, 35, 42, 43,
                               60,
                               ...,
                               21769, 21771, 21774, 21782, 21800, 21801, 21811, 21812, 21813,
                               21820],
                              dtype='int64', length=3863),
 'Botafogo': Int64Index([ 23, 48, 87, 88, 111, 119, 127, 134, 196,
                          200,
                          ...,
                          21471, 21487, 21499, 21529, 21653, 21660, 21666, 21715, 21746,
                          21790],
                         dtype='int64', length=873),
 'Copacabana': Int64Index([ 0, 9, 10, 11, 24, 25, 28, 31, 86,
                           91,
                           ...,
                           21707, 21713, 21736, 21743, 21780, 21783, 21795, 21809, 21810,
                           21815],
                          dtype='int64', length=2644),
 'Flamengo': Int64Index([ 78, 138, 218, 284, 321, 347, 356, 361, 369,
                           393,
                           ...,
                           21527, 21560, 21581, 21629, 21680, 21704, 21728, 21731, 21740,
                           21794],
                          dtype='int64', length=714),
 'Ipanema': Int64Index([ 19, 39, 40, 45, 52, 53, 73, 84, 92,
                         114,
                         ...,
                         21718, 21722, 21730, 21739, 21744, 21745, 21754, 21788, 21796,
                         21817],
                        dtype='int64', length=1764),
 'Leblon': Int64Index([ 27, 59, 90, 97, 115, 133, 161, 167, 171,
                        213,
                        ...,
                        21721, 21761, 21770, 21777, 21785, 21786, 21787, 21808, 21816,
                        21825],
                       dtype='int64', length=1258),
 'Tijuca': Int64Index([ 15, 18, 20, 80, 82, 126, 145, 149, 158,
                        164,
                        ...,
                        21573, 21577, 21607, 21610, 21661, 21675, 21714, 21729, 21773,
                        21818],
                       dtype='int64', length=1100)}
```

In [10]: `# visualizando a média para cada bairro selecionado
grupo_bairro['Valor'].mean()`

Out[10]:

```
Bairro
Barra da Tijuca    7069.552938
Botafogo           8791.828179
Copacabana         4126.677005
Flamengo           4113.526611
Ipanema            9352.001134
Leblon             8746.344992
Tijuca             2043.520000
Name: Valor, dtype: float64
```

In [11]: `# passando mais de uma variável neste espaço,
como 'Valor' e 'Condominio', e utilizando o método
round() para arredondar os valores
grupo_bairro[['Valor', 'Condominio']].mean().round(2)`

Out[11]:

	Valor	Condominio
Bairro		
Barra da Tijuca	7069.55	3591.01
Botafogo	8791.83	976.28
Copacabana	4126.68	1148.68
Flamengo	4113.53	1102.15
Ipanema	9352.00	2244.44
Leblon	8746.34	2107.18
Tijuca	2043.52	711.69

Estatísticas Descritivas

In [12]: `# visualizando o grupo de bairros
método describe() que gera estatísticas descritivas
round() para arredondar os valores
grupo_bairro['Valor'].describe().round(2)`

Out[12]:

	count	mean	std	min	25%	50%	75%	max
Bairro								
Barra da Tijuca	3863.0	7069.55	11874.15	800.0	2500.0	4500.0	8500.0	600000.0
Botafogo	873.0	8791.83	152202.41	700.0	2200.0	3000.0	4350.0	4500000.0
Copacabana	2644.0	4126.68	3611.41	100.0	2000.0	3000.0	4800.0	35000.0
Flamengo	714.0	4113.53	3839.13	800.0	1900.0	2900.0	4975.0	35000.0
Ipanema	1764.0	9352.00	8219.72	1200.0	4500.0	7000.0	11000.0	90000.0
Leblon	1258.0	8746.34	7004.04	100.0	4500.0	7000.0	10500.0	100000.0
Tijuca	1100.0	2043.52	1664.34	750.0	1500.0	1800.0	2300.0	45000.0

Obs.: podemos perceber alguns problemas: como o desvio padrão(std) apresenta uma variação muito grande no caso de Botafogo, o mesmo ocorre para Barra da Tijuca. No caso de valor máximo, podemos perceber valores estranhos também; segundo a tabela, existem imóveis disponíveis em em Botafogo com a locação de 4500000. Essas informações estão errôneas, muito provavelmente esses valores correspondem à venda e não à locação.

In [13]: `# recebendo os conjuntos estatísticos que nos interessam
grupo_bairro['Valor'].aggregate(['min', 'max']).rename(columns = {'min': 'Mínimo', 'max': 'Máximo'})`

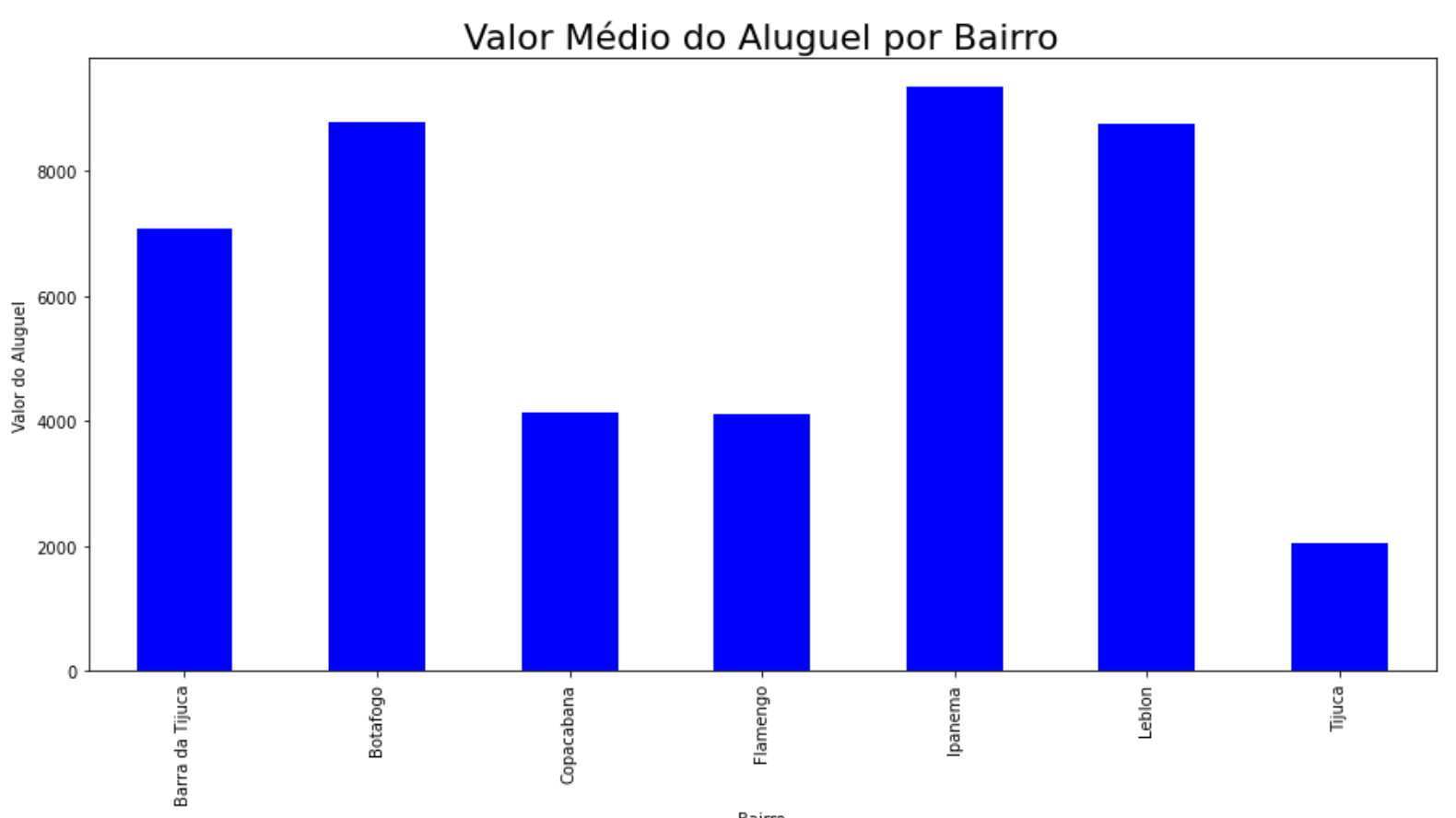
Out[13]:

	Mínimo	Máximo
Bairro		
Barra da Tijuca	800.0	600000.0
Botafogo	700.0	4500000.0
Copacabana	100.0	35000.0
Flamengo	800.0	35000.0
Ipanema	1200.0	90000.0
Leblon	100.0	100000.0
Tijuca	750.0	45000.0

In [14]: `# importando Matplotlib lib
para gerar visualizações gráficas
configurando o tamanho da representação visual via figsize()
%matplotlib inline
import matplotlib.pyplot as plt
plt.rc('figure', figsize = (15, 7))`

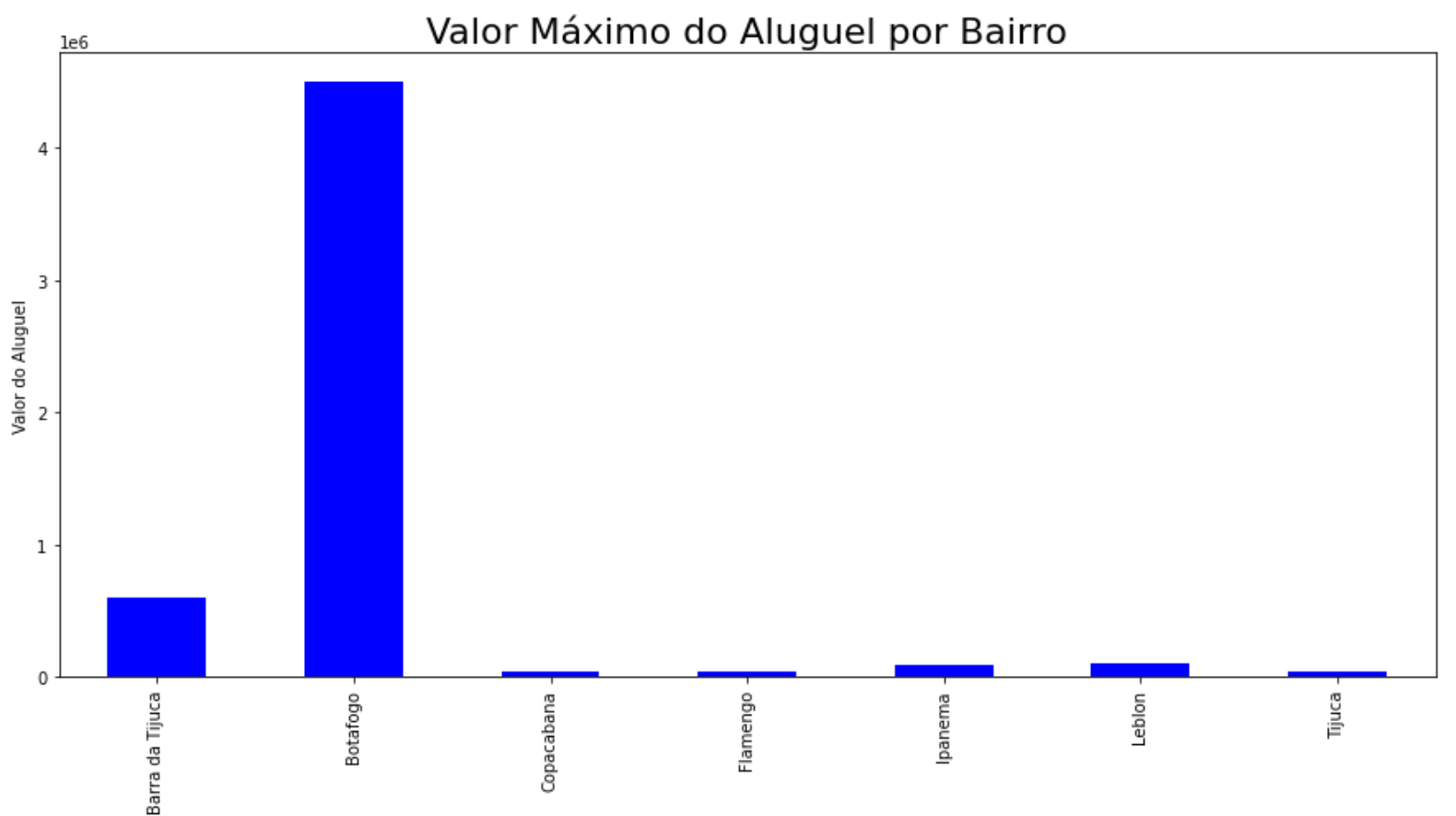
In [15]: `# construindo primeiro plot com 'grupo_bairro', e atribuindo essa representação à variável f
fig
modificando o index do eixo Y
adicionando um título ao gráfico
fig = grupo_bairro['Valor'].max().plot.bar(color = 'blue')
fig.set_ylabel('Valor do Aluguel')
fig.set_title('Valor Médio do Aluguel por Bairro', {'fontsize': 22})`

Out[15]: Text(0.5, 1.0, 'Valor Médio do Aluguel por Bairro')



In [16]: `fig = grupo_bairro['Valor'].max().plot.bar(color = 'blue')
fig.set_ylabel('Valor do Aluguel')
fig.set_title('Valor Máximo do Aluguel por Bairro', {'fontsize': 22})`

Out[16]: Text(0.5, 1.0, 'Valor Máximo do Aluguel por Bairro')



In [17]: `fig = grupo_bairro['Valor'].min().plot.bar(color = 'blue')
fig.set_ylabel('Valor do Aluguel')
fig.set_title('Valor Mínimo do Aluguel por Bairro', {'fontsize': 22})`

Out[17]: Text(0.5, 1.0, 'Valor Mínimo do Aluguel por Bairro')

